# HP Professional

THE MAGAZINE FOR THE BUSINESS & PRACTICE OF HEWLETT-PACKARD COMPUTING

JULY 1987 ■ VOL. 1, NO. 2

- UNIX For Real-Time Applications
- The Truth About IMAGE
- A Bold Approach To Spectrum


**HP 3000**

*D*

*M*

FOCUS

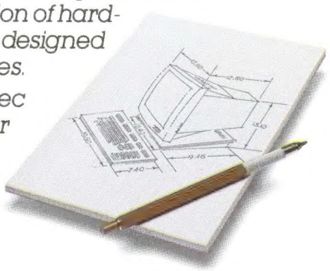
CIM/FACTORY AUTOMATION


## You imagined it. We created it. The Zentec 8392 Terminal.

*When it comes to finding an ideal partner, it's important to do a little comparison shopping—especially if your current partner operates under the name of HP 2392A.*

The Zentec 8392 offers everything the 2392A user needs and more. For less. Like a 14-inch diagonal screen. Yet hidden behind its sleek, sophisticated exterior lies a powerful collection of hardware and software specifically designed to outperform other compatibles.

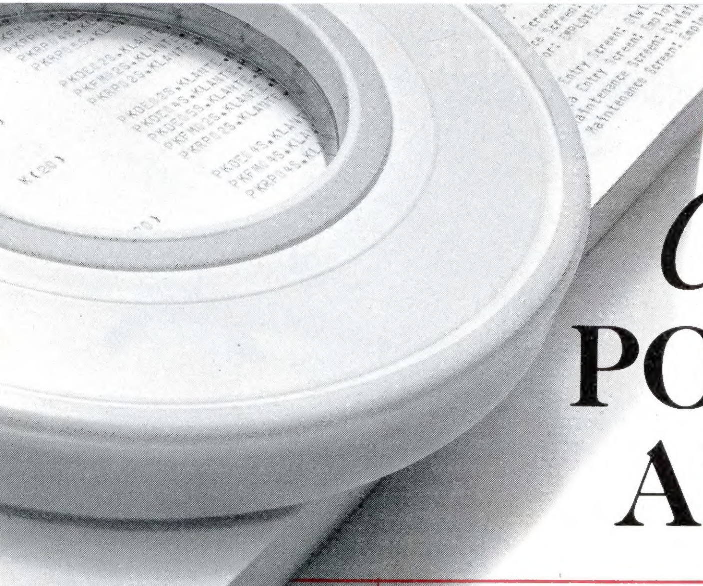
When you work with the Zentec 8392, we promise you a partner who operates in both 80 and 132 columns with a memory better than your own (up to 8 pages). This partner's bright (display available in green or amber), flexible and reliable. Plus, there's a one-year warranty and maintenance contracts available through Dow Jones Service Company.



So if you've been playing the silent partner, merely taking what you get because it's there, shop around. Look at Zentec. Once you do, you won't want to look back. For more information, call 408-727-7662 (inside California), 800-332-5631 (outside California), or 011-44-4867-80666 (in the UK). Zentec Corporation, 2400 Walsh Avenue, Santa Clara, CA 95051-1374.

ZENTEC™
THE CUSTOM TERMINALS COMPANY

ENTER 111 ON READER CARD



FOR USERS OF
POWERHOUSE ON HP 3000

# *Cognos Introduces* POWERHOUSE® ARCHITECT

## Effective Documentation Makes All the Difference!

If you're spending too much time documenting your applications or don't have enough time to document completely, then POWERHOUSE ARCHITECT is just what you need.

This new option to POWERHOUSE lets you concentrate on application development and forget about documentation.

## Automatic Documentation for Standardization and Consistency

Get three levels of automatic documentation for new and existing POWERHOUSE applications. Prototype, for a basic description of the system you're testing. User, for a more complete set of instructions to accompany the production version of the application. And System, for a detailed technical document describing all aspects of your finished application.

Each level uses information taken directly from the source code. The result is standardization and consistency of documentation. This also means improved maintenance and less time required for a new user to learn the application.

## On-Line Cross Referencing for All Affected Programs

When making modifications to the data structure for an application designed with POWERHOUSE ARCHITECT or an existing application loaded into POWERHOUSE ARCHITECT, you can now automatically generate a complete list of all programs affected by a specific change. Not only can files and elements be cross-referenced to programs, but also subfiles, temporary items, defined items and designer procedures.

> "The ARCHITECT computerizes the work of computer programmers. It provides the quantum leap in productivity for MIS personnel that computerization has provided for other departments and operations."
>
> Alan Bose, Manager, Information Services, Continental Web, Itasca, Illinois.

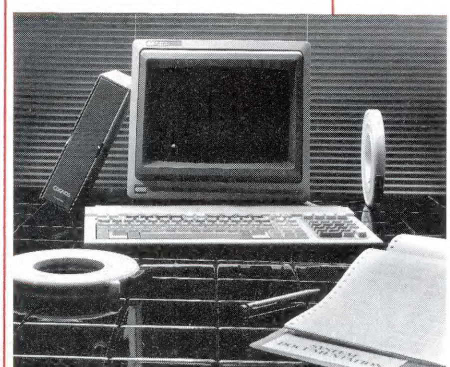
## On-Line Dictionary Faster and Easier

Building a dictionary is now even faster and easier — especially for those using POWERHOUSE for the first time.

## Automatic Default Prototype Generator

Once your new dictionary is created, POWERHOUSE ARCHITECT can have your working prototype up and running in minutes.

POWERHOUSE ARCHITECT is a whole new approach towards easy and effective application development. It's your blueprint for building better applications productively.



COGNOS

ENTER 101 ON READER CARD

*For complete information on* **POWERHOUSE ARCHITECT** *call now!*

**In the U.S. call 1-800-4-COGNOS • In Canada call 1-613-738-1440 • In the U.K. call +44 344 486668**

2 Corporate Place, I-95, Peabody, MA. 01960

# THE DESIGNER PRINTER.


## TDP and the LaserJet with PSP/Plus.

If you use TDP, you probably fall into one of two classes: you either have a laser printer, or you want one. If you already have a laser printer, you're probably used to producing high-quality illustrated documents at a whim. If you don't, you're probably working with a daisy-wheel printer regarded by its users as a faithful but aging pet spaniel.

PSP/Plus and the LaserJet can provide you with all the benefits of laser printing at about a quarter the price of the HP2688: full graphics capability, up to 31 fonts in a document, forms, borders, logos — in fact, almost everything that TDP can make a big laser printer do. It works with popular "front end" processors, too, as well as the TDP Mailer facility. In fact, with PSP/Plus, the LaserJet can be used with your own applications and with HP software: HPDraw, DSG, HPEasyChart, HPWord, IDS and IFS all work with the LaserJet, thanks to PSP/Plus.

So if you want to replace the aging daisy-wheel printer, or get to all the power that TDP *really* has, or just want to avoid walking four blocks to the data center's HP2680, try a LaserJet: with PSP/Plus, the LaserJet has joined the HP3000 family.

299 W. Foothill Blvd. • Suite 230 • Upland, CA 91786
714/985-1581 • 800/858-4507 • Telex 532678 OPT INC

**ENTER 113 ON READER CARD**

HP2680, HP2688, TDP, HPWord, DSG, HPEasyChart, IDS, IFS, and the LaserJet are products of Hewlett-Packard Company.

OPT

# CONTENTS

JULY 1987 VOL. 1, NO. 2



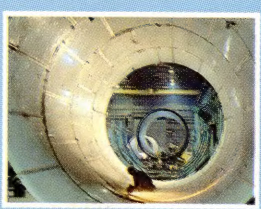
## COLUMNS



## DEPARTMENTS



## COVER



*This month's cover photo shows a worker at the Paul Mueller Company (Springfield, MO) grinding and finishing out weld seams on large beer storage tanks.*

*Preliminary engineering work for these tanks is done on a Hewlett-Packard DesignCenter Mechanical Engineering Series 10 running on an HP Series 9000 Model 320 workstation.*

**FOCUS**
CIM/Factory Automation
26

# HP Professional

## Editorial

**MANAGING EDITOR** Lonni Wright
**COPY EDITOR** Karen Detwiler
**CONTRIBUTING EDITORS** Jim Dowling, Don Person
**CONTRIBUTORS** Michael A. Casteel, Suzanne M. Doughty, Robert Green, David Greer, Karen Heater, Sol F. Kavy, Steven R. Kusmer, Douglas V. Larson, David Merit, Lynn Parker, Michael Shumko

## Design

**DESIGN/PRODUCTION DIRECTOR** Leslie A. Caruso
**DESIGN/PRODUCTION ASSOC.** Ruth Ann Leiby
**ART/PRODUCTION ASSOC.** Timothy M. Kraft
**ART ASSISTANT** Sue Ann Rainey
**PRODUCTION ARTIST** Richard Kortz
**SENIOR TYPESETTER** Joseph E. Hohenwarter
**TYPESETTING/PRODUCTION** MaryEllen Springer
**DESIGN CONSULTANT** Greg Paul, Brady & Paul Communications

## Circulation & Administration

**VICE PRESIDENT** Peg Leiby
**CIRCULATION DIRECTOR** Mary Wardlaw
**CIRCULATION MANAGER** Karen Savory
**CIRCULATION FULFILLMENT** Virginia Ashabranner, Josh Friedman
**ACCOUNTING** Andrea Beneke
**COMPUTER SYSTEMS** Bonnie Auclair, Kevin Kennelly, Ruth Mermelstein

**PROFESSIONAL PRESS, INC.**
Publisher: Carl B. Marbach
Editorial Director: R. D. Mallery
Vice President: Peg Leiby
Vice President: Helen Marbach

For information on how to contact your sales representative, see page 98. Editorial, advertising sales and executive offices at 921 Bethlehem Pike, Spring House, PA 19477 ■ (215) 542-7008 TWX 910 333 9522 ■ Easylink 62805174

HP PROFESSIONAL is published monthly by Professional Press, Inc., 921 Bethlehem Pike, Spring House PA 19477. Subscriptions are complimentary for qualified U.S. and Canadian sites. Single copy price, including postage, $4. One year subscription rate $30 U.S. and Canada; $60 foreign. All orders must be prepaid. Second class permit applied for and pending at North Wales, PA and additional entry. POSTMASTER: Send all correspondence and address changes to HP PROFESSIONAL, P.O. Box 445, Spring House, PA 19477.



**BPA Membership applied for 4/87.**

# On The Road

The success of a magazine is measured in many ways. Advertising sales and the number of new subscribers provide a publisher with valuable information about the future of a publication. They are the financial backbone of any magazine, providing its staying power. These figures have been very encouraging after the release of the premier issue of *HP PROFESSIONAL.*

But from an editor's point of view, feedback from HP users, professional writers and Hewlett-Packard employees who want to participate in the magazine is invaluable. And we've heard from you about *HP PRO*! You're interested and you want more.

It's exciting to talk with people who are enthusiastic about a new publication. You want to publish articles, some of you for the first time. And you offer constructive criticism, new ideas and approaches, and interesting ideas for future issues. Your comments tell us that *HP PRO* is on the road to success.

Because that's what it's all about with a magazine like this. And as long as you want to be a part of it, it will be around.

An important objective of ours is to cover both commercial and technical aspects of the HP world, with the intention of making *everyone* understand the technology behind 3000s, 9000s, 1000s — to see HP from all perspectives.

An HP 3000 professional, for example, may never work on a 9000 system, but if he understands the capabilities and limitations of the 9000, he'll have a better understanding of the HP universe. So we'll publish articles on all aspects of HP technology.

With that aspiration, we'll make the HP community stronger and *HP PROFESSIONAL* will be more valuable.

We have a lot planned for the remainder of this year. We'll continue our coverage of Spectrum developments and provide third-party vendors' experiences in migrating to the new systems (see "Infocentre Takes Different Approach To Spectrum," this issue).

Each issue will bring you the best possible coverage of Hewlett-Packard news and product announcements, as well as announcements from third-party vendors.

Our featured topics planned for the rest of '87 include integrated business systems, DBMS, graphics, artificial intelligence, systems performance and languages.

We'll also be sure to bring you other newsworthy topics like desktop publishing, peripherals, office automation, security and more.

Keep letting us know what you want to read — and write. Contribute your feature articles, technical tips, news stories, software and hardware reviews, calendar events, product announcements, letters, editorials . . . whatever you're inspired to do. We take our direction from you and will go where you lead us.

We're looking forward to hearing from you soon . . . and often!

Lonni Wright

# QUESTIONS & ANSWERS

*Editor's Note: This month Jim Dowling deals with a group of questions concerning "spooling" and printers.*

*If you have questions concerning any aspect of Hewlett-Packard computer operation and applications, send them to Q & A, HP PROFESSIONAL, P.O. Box 445, Spring House, PA 19477.*

## SPOOLING AROUND

**QUESTION:** *How do I configure the "MAX # OF OPENED SPOOLFILES" value in the configuration dialogue?*

**ANSWER:** The penalty for having this parameter set too low will be a severe case of SJDS (Sudden Job Death Syndrome) or SPDS (Sudden Program Death Syndrome). As a job attempts to log on, it will allocate a spooled file for its input ($STDIN) and one for its output ($STDLIST). If the current number of opened spooled files exceeds the maximum configured value, the job will abort and a message will be displayed on the system console. If a program fails to open a spooled file for this reason, the program will abort.

The maximum allowed value depends on the operation system version that you're running. The limit for E.00.00 or F.00.00 base MITS is 255 and for G.00.00 is 1023. Each executing job requires two plus the number of simultaneously opened print files that will occur in the job. In addition, sessions that require spooled print files will add to the number. A rule of thumb would be to use 3 * (#JOBS + #SESSIONS). With a job limit of 10 and a session limit of 60, max spool files should be set to 210.

## ACCOUNTS RECEIVABLE

**QUESTION:** *My program blows up whenever I try to print the entire General Ledger Account Structure with details of the year's transactions. How can I reconfigure my system to handle this print file?*

**ANSWER:** There are two causes for this type of program failure that can be attributed to system configuration. Either the print file filled all usable disk space or it was so large that it took more than the maximum allowable number of noncontiguous disk areas to create the file.

All spooled files are stored on disk on their way to their final destination. Only disks that have the Device Class Name of SPOOL will be used. If there is insufficient available space, the program will fail. Be sure to configure as many drives as possible with the class name of SPOOL and manage available disk space to ensure successful spooler operation.

When a spooled file is created, it will be built as a series of disk space areas called Extents. The size of each extent is determined by the value set in response to the System Configurator questions: "# OF SECTORS PER SPOOLFILE EXTENT." The MPE File system allows a maximum of 32 extents per file; therefore, the upper limit of disk sectors for a print file will be 32 times the configured extent size. Although spooled print fields are generally of undefined length, an extent size of 1024 sectors should support a minimum print file size of 1250 pages. Increasing the extent size proportionally increases the maximum number of printed pages that the spooler will support.

There are two additional factors that come into play. The configuration parameter "MAX # OF SPOOLFILE KILOSECTORS" establishes an upper limit on the amount of disk space that will be allocated for all spooled files. You can exceed this value with less than 32 extents and before filling the disks thereby causing a program or job failure. The second factor is disk fragmentation. As files are built and purged from the disks, the disk space becomes scattered with intermixed available and used regions. Each extent of a spooled file must be placed in a contiguous free space region. If no regions of sufficient size exist, the effect will be the same as if no disk space were available or more than 32 extents were required.

## FRAGMENTATION

**QUESTION:** *Often when my dissertation on spooling is done, someone asks, "What if I keep getting killed by fragmentation?"*

**ANSWER:** I'd suggest that the system be configured to handle print files of about 250 pages and alter your program to keep track of its current page number, rather than close and reopen the print file after every 500 pages.

## DOUBLE WHAMMY

**QUESTION:** *At this point the group gangs up on me with two simultaneous comments/questions: "I don't have the source code" and "What if the print file is bigger than my total disk free space?"*

**ANSWER:** Wow! There are two general approaches to this. The quick fix would be to stop spooling on the print and run

"What if..."

# Announcing a great way to power up your HP 3000!

**Purchase a board-swap upgrade by August 31 and get an HP Vectra 3000 PC and valuable PC software at no additional cost.**
Starting today, upgrading your HP 3000 computer is more rewarding than ever before.
You see, with just a simple and affordable board-swap upgrade, your present HP 3000 can become dramatically more powerful. That means you'll get a faster CPU which can boost performance by 20 to 50%. Your new system will also support up to 4 times the main memory capacity you've got now, and we're including 4 megabytes, standard, in most cases. Since the entire process takes only a few hours, there is minimal disruption. And no retraining sessions.

**The reward?** Shorter response time, happier users, and increased productivity, all at prices that are up to 20% lower than last year. Your upgraded system can also support additional users, applications, and outstanding Hewlett-Packard peripherals.
**Extra reward:** If you purchase a board-swap upgrade for your Series 39, 40, 42, 44, 48, 64 or 68 by August 31, you'll receive an IBM PC/AT compatible Vectra 3000 PC and the Vectra Office Professional Software Pack. All for the price (before purchase agreement discounts) of the upgrade alone. What a great chance to try out our powerful productivity software on your own HP Vectra 3000 PC, in your own work environment.
**Reward yourself:** Find out how to expand the power of your HP 3000 (and get a color Vectra 3000 PC and productivity software when you upgrade). Call Hewlett-Packard now. It's the perfect time for your company to power up!

**Call toll free: 1-800-558-3077, ext. 3001.**

ENTER 191 ON READER CARD

HEWLETT PACKARD

ISO 2721

the program to the printer "hot." But no other process can access that printer until the report is completed and spooling is restarted. The result is essentially a down system for the duration of a very slow process. Remember, the program is now limited by the speed of the printer, not the speed of the spooling disk.

The second approach would be to route the print file to a magnetic tape, then copy the tape to the spooled printer in small chunks. Either a utility program could be written or FCOPY could be used with its SUBSETing options.

## COOL START

**QUESTION:** *Why do spooled print files go away when the system is COOL started?*

**ANSWER:** Spooled print files are not documented in the system file directory like permanent files are. A special table called the Output Device Directory is used to keep track of them. When the system is WARM started, the memory resident table is retained; when COOL started, this table is rebuilt. And by the way, the disk space used by spooled files before the COOL start or even COLD load will be lost from the system free space tables until it's recovered either by a RELOAD or executing a RECOVER LOST DISK space at startup.

## SETUP SETBACK

**QUESTION:** *We set up a Microfiche procedure with a service bureau that seemed to work fine until we sent them a tape created by a different program. The vendor says that the carriage control is different and another setup charge will be required. What happened? Can this be avoided?*

**ANSWER:** The problem most likely resulted from the file system feature that allows both prespacing and post-spacing when writing carriage control. Prespacing means "advance lines then print" whereas post-spacing means "print this then advance lines." (Do not confuse this with "BEFORE ADVANCING" or "AFTER ADVANCING" in COBOL). The two modes even can be mixed in the same print file. Since each program could set up one or the other of these spacing modes, you should look into your print files using the SPOOK utility. SPOOK has a control directive that will allow you to see the carriage controls in use as you list the file. The command "MODE CONTROLS=ON" will show you what's going on.

To avoid the problem and to allow you to put many microfiche jobs on one tape, you could use a report header standard as follows: At the top left corner of each page, place a constant string of characters, say your company name. On the next line and directly below it, place the report identifier, say a report number. Below that, place an indexing tag for the report page.

Now, the fiche generation program can be set as follows: Perform a frame break every time the company name appears starting in column two (not one; we will ignore the carriage control directives that the print would use); index the frame from line three below; and perform a fiche break every time the report identifier on line two changes. This works quite effectively allowing you to put as many different reports as you choose onto a single tape. You can fix the locations of RUN-DATE, REPORT TITLE and other fiche header data to complete the setup. Keep your tape file block size within the limits of the machine that will be generating your fiche; other lines may be lost.

## THREE ROUTES

**QUESTION:** *Every once in a while, I would like to route something to the line printer that a program displays on my terminal. How can I redirect output if I don't know what it's writing to?*

**ANSWER:** Several possible solutions are available. The most general is to redirect $STDLIST which is where the program displays go from your terminal screen to the printer. Try "FILE OUT;DEV=LP" then ":RUN PROGRAM;STDLIST=OUT". This should do the trick for many programs.

Another possibility would be to hit the BREAK key while the program is listing to the terminal, then type "SHOWOUT". This may reveal the name of the file that the program has routed to your terminal display. If it does, try issuing a file equation to redirect it. This will not always work because some programmers will "disallow file equations" for that file.

A third option is available for routing MPE command displays to the printer: Execute a program that allows the invocation of MPE commands from within it. FCOPY will work fine if you do the following: "FILE OUT;DEV=LP"; then "RUN FCOPY.PUB.SYS;STDLIST=OUT"; then from within FCOPY, type ":SHOWME" or whatever MPE command you choose. The display (and all FCOPY prompts) will be sent to the printer.

## GREAT EXPECTATIONS

**QUESTION:** *Why does my eight-page per minute LaserJet take two minutes to print a page?*

**ANSWER:** This problem is actually a case of unreasonable expectations. The LaserJet printers will print at a rate of eight pages per minute once the entire page has been loaded into its memory. Only if you use the multi-copy feature of the LaserJet should you expect to see eight pages come out in a minute. A complex graphics image will take quite a long time to download. The exact time required to transfer a page is dependent on data transfer speed and the number of bytes necessary to define the page. The same situation exists for the HP 2680 and HP 2688 laser printers.

To wrap up this month's installment, I'd like to share a few things from the May NECRUG meeting in Atlantic City. I came across a company that sells inexpensive ($1,200 a piece) 2400 bps modems that have a data compression feature. This allows you to drive a 300 lpm printer at full throttle across a 2400 bps line. I'll provide more on this next month. Also, Valdimir Volokh reminded me that VESOFT's MPEX program can be used to PURGE spoolfiles without losing disk space and that it's much quicker than using SPOOK. ■

# Precision Architecture Family Expanded By Three

*Model 840 Also Enhanced*

HP's latest announcement of RISC-based computers includes three new advanced computer systems, ranging from a high-performance superworkstation to a new top-of-the-line superminicomputer, for engineering, manufacturing and scientific computing applications.

All three systems employ the company's Precision Architecture design, which gives the new machines excellent price/performance and reliability characteristics compared with competitive systems from other major vendors.

The company also announced performance enhancements to its original HP Precision Architecture machine, the HP 9000 Model 840, which was first shipped in November 1986.

The new HP 9000 Model 825SRX, Model 825S and Model 850 use the HP-UX operating system.

The new systems are said to offer substantially better price/performance than comparable systems from Digital Equipment Corporation, largely because of inherent product economies made possible by the simplified design. For example, the Model 825S has two to three times the performance of DEC's VAX 8250 at about two-thirds the price, and about one-third higher performance than the VAX 8350 at half the price, or at about the price of a MICROVAX II.

The new systems are the first Series 800 machines to incorporate HP's NMOS III VLSI chip technology. First implemented in 1982 on the HP 9000 Model 500 workstation, the technology enables electronic circuitry to be packed with extreme density, which contributes to the excellent price/performance characteristics of all three machines.

The new Model 850S becomes the top-of- the-line Precision Architecture system for technical applications. It is comparable in performance to DEC's VAX 8550 and 8700 computers for as little as half the price. Model 850S performance is seven million instructions per second (MIPS) in a multiuser environment.

The superminicomputer will support 50 to 60 active users at introduction, with a maximum of 90 terminal connections. Significantly higher multiuser capacity is planned for the future. Maximum memory capacity is 128 MB and disk-memory capacity is 6.85 gigabytes.

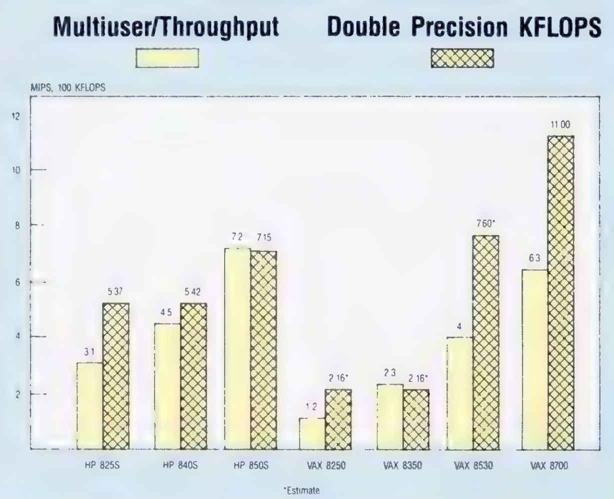
Typical 850S applications include:

- *Simulation and modeling in research laboratories.*
- *Packet-switching and network management and control in telecommunications industries.*
- *Topological mapping and seismic analysis in the earth-resources area.*
- *Real-time data acquisition and control in discrete manufacturing and process industries.*

Priced at $200,000, the Model 850S system-processing unit includes a CPU, power supply, 16 MB of error-correcting memory, HP-IB parallel interface, six-channel serial multiplexer, 32-user HP-UX license and 10 I/O slots for connections to terminals, disk drives, tape drives, printers, instruments and networking devices. Delivery is estimated at 24 weeks after receipt of order.

The Model 825S becomes the lowest-priced, entry-level system in the Precision Architecture family. It serves two to 24 active users and supports a maximum of 66 terminal connec-

**HP vs. DEC Performance Comparisons**



tions. It has a main-memory capacity of 56 MB and a disk-memory capacity of 6.85 gigabytes.

The Model 825S performs at three MIPS in a multiuser environment. It is intended for use as a general-purpose computer in a UNIX operating system environment, as a server in a small network of engineering workstations and as a workcell controller or area manager in industrial-automation applications.

Manufacturing applications for the Model 825S include real-time data acquisition and control, quality management, dynamic scheduling, computer-aided process planning, process optimization and simulation, distributed numerical control and discrete-manufacturing monitoring and control.

The Model 825S system-processing unit at $42,500 includes a CPU, floating-point processor, power supply, 8 MB of error-correcting memory, HP-IB parallel interface, six-channel multiplexer, 16-user HP-UX license and five available I/O slots for peripheral devices, instruments and local area network connections. Delivery is estimated at 16 weeks after receipt of order.

The Model 825SRX superworkstation is believed to deliver more performance than any other workstation available. Its computational performance and high-speed bit-mapped graphics place the HP superworkstation at the top of this new class of products.

The superworkstation's performance is eight MIPS in a workstation environment. Relative to DEC's VAX 11/780 integer benchmarks, Model 825SRX has 8.2 times the performance, according to HP. It supports up to 48 MB of error-correcting RAM and delivers more than twice the system throughput of the Model 350.

The Model 825SRX will be used primarily by mechanical engineers in 3D solid modeling for applications such as conceptual design, visualization, analysis, testing and finite-element modeling and analysis. Other applications that require the product features of Model 825SRX include general graphics, real-time/simulation, mapping, modeling and image processing.

It is object-code compatible with all Series 800 products and source-code compatible with all Series 300 products. This provides a uniform program-execution environment across the HP 9000 product family.

The base configuration of the superworkstation at $69,500 includes the system processing unit, 8 MB of RAM, HP-IB interface, keyboard, mouse, HP-UX, Starbase graphics, ARPA and BSD 4.2 networking services, Ethernet/IEEE 802.3 LAN interface and the SRX graphics package. It includes the display controller, graphics accelerator, eight-plane frame buffer, and 19-inch high-resolution color display (1,280 X 1,024 pixels). Delivery is estimated at 16 weeks after receipt of order.

*Solid objects can be rendered at interactive speeds with the HP 9000 Model 825SRX, HP's new superworkstation.*

HP has increased the maximum memory capacity and the number of active users supported on its first Precision Architecture system, the model 840S minicomputer.

Maximum memory is increased from 24 MB to 96 MB. By increasing the number of I/O slots from 12 to 28, Model 840S now can support from 16 to 50 active users, and a maximum of 128 terminal connections.

The enhanced Model 840S delivers 4.5 MIPS performance, comparable to the DEC's VAX 8530 and VAX 8600 systems, but at about half the price.

The Model 840S is priced at $81,500 and comes standard with 8 MB memory, floating-point coprocessor, disk interface, six-channel mux, access port card and 16-user HP-UX license (with C compiler, assembler, symbolic debugger, Port/HP-UX and real-time extensions) and is available for immediate shipment.

Because all HP 9000 computers use the HP-UX operating system, users benefit from a high degree of compatibility across the entire line, as well as compatibility with programs and applications that are developed under UNIX operating systems on other vendors' computers. HP-UX has passed compliance tests imposed by the System V Verification Suite (SVVS).

Communication among HP 9000 computers and other vendors' computers is made possible by the de facto standard U.S. Department of Defense's Advanced Research Projects Agency (ARPA) and BSD 4.2 networking services, included with all new models. HP's Network Services also are available on all new models, providing a

| HP 9000 Series 800 Multiuser System Specifications | | | |
|---|---|---|---|
| | 825S | 840S | 850S |
| MIPS | 3.1 | 4.5 | 7.2 |
| Floating Point<br>Whetstones (SP)<br>Whetstones (DP)<br>Linpacks (SP)<br>Linpacks (DP) | <br>3521<br>2433<br>680<br>537 | <br>3115<br>2092<br>738<br>542 | <br>4202<br>2907<br>869<br>715 |
| Memory (Std./Max.) | 8/56 MB | 8/96 MB | 16/128 MB |
| I/O Slots (Std./Max.) | 7/12 | 12/26 | 18 |
| HP-UX License | 16 | 16 | 32 |
| SPU Size (sq. cm.) | 1625 | 5700 | 9100 |
| Operating Temp. Range | 0-55 C | 0-55 C | 5-40 C |
| SPU Price | $42,500 | $81,500 | $200,000 |

distributed-file system and communications with HP 1000, HP 3000 and DEC VAX computers.

IBM mainframe communications is provided by HP's SNA 3270 for 3278 terminal access and PC3270 file-transfer capabilities.

X Window, the de facto windowing standard, also is available on the new computers. X Window is a set of primitives (library of subroutines) that supports the development of windowing application. X Window provides high-performance, high-level, device-independent graphics and allows the new models to work in a multivendor, networked environment.

VLSI chip design in HP Precision Architecture products has resulted in computers with fewer parts, higher reliability and lower monthly-support costs. Series 800 computers also provide higher overall environmental tolerance than many other minicomputers, which reduces on-site requirements and costs.

## DBGENERAL Goes To Olympic Festival

*Chosen Official Database Utility*

DBGENERAL, the IMAGE and TurboIMAGE database utility from Bradmark Computer Systems (Los Angeles, CA), has been selected the official database utility of the U.S. Olympic Festival, sponsored by North Carolina Amateur Sports (NCAS).

DBGENERAL will be used to restructure, maintain, diagnose and correct problems in NCAS's databases on an HP-donated Series 70.

## HP Signs Contract For QMS Controllers

*Companies Form Strategic Alliance*

QMS, Inc. (Mobile, AL), recently signed an OEM contract with Hewlett-Packard for QMS MAGNUM printer controllers.

Under the agreement, HP will use the MAGNUM controllers in its complete line of dot matrix impact printers. QMS also will be developing new, enhanced MAGNUM products.

MAGNUM controllers, which use the proprietary QMS MAGNUM programming language, have become de facto standards for industrial graphics printing.

Because of the controllers' advanced plotting capabilities, they are well suited to bar code and label printing applications in such industries as automotive, retail and health care.

## NCGA Calls For Animation Entries

*Awards Given At NCGA '88*

The National Computer Graphics Association (NCGA) is calling for entries in its competition designed to recognize the best in computer animation.

The 1988 International Computer Animation Competition is open to all individuals and organizations engaged in producing computer animation. This is the third year NCGA has sponsored the competition.

Entries will be judged in 11 categories, both professional and non-professional, on the creative uses of computer graphics and technical quality. The categories are as follows:

Professional — broadcast computer graphics, corporate logos, television commercials, corporate communication computer graphics, music videos, research computer graphics, science and industry computer graphics, theatrical motion picture graphics, non-commercial films.

Non-professional — secondary/undergraduate, graduate/faculty.

All first-place winners will receive their awards at the NCGA Awards and VideoGala dinner, March 22, held during NCGA '88, March 20-24, 1988, in Anaheim, CA. Winning entries will be screened during the reception and dinner.

To be considered for this year's competition, entries must be completed after January 1, 1987, and received at NCGA headquarters no later than 5:00 p.m. EST on December 1, 1987.

A call for entries brochure and additional information can be obtained from NCGA's communications department, (703) 698-9600.

## Series 950 Introduced For Late 1987 Deliveries

*Series 930 To Ship In August*

Initial shipments of the HP 3000 Series 930 and Series 950 computers remain on schedule for August and the fourth quarter of 1987, respectively.

In making announcements concerning the first business computers to employ the RISC-based HP Precision Architecture design, HP also placed the Series 950 — its 7-MIPS, top-of-the-line business system — in active order status and announced a new price for the Series 930.

According to Douglas C. Chance, HP senior vice president and general manager of HP's Business Systems Sector, the company has discovered no major difficulties that would cause a revision of the shipping schedule. The first shipments of the Series 930 will begin in August, within the mid-1987 time frame the company projected last September. The Series 950 should begin shipping in the fourth quarter of 1987.

Chance also said the company's ongoing effort to tune the MPE XL operating system "continues to show good results," though performance tuning will continue throughout the Beta-testing process.

The new U.S. list price for the Series 930 — $180,000 for a base system, down from $225,000 — is a response to new competitive offerings since the systems were announced in February 1986.

According to Chance, simplicity of design and lower part counts resulting from Precision Architecture make it possible for the company to make this reduction.

***The HP 3000 family. Front: Series 950 (left) and 930. Back: Micro 3000 (left), Series 52, Series 70 and Series 58.***

Series 950, the most powerful business system HP has ever offered, is list priced at $260,000. It can be ordered now.

Both systems are priced 20 to 50 percent lower than comparable processors from IBM and Digital Equipment Corporation.

Series 950 is implemented using advanced NMOS III integrated-circuit technology. When combined with HP Precision Architecture, this technology gives Series 950 cost-effective, reliable, high performance. The entire CPU of Series 950 is contained on a single VLSI chip, and all process circuitry is on a single printed-circuit board (including the CPU chip, 128 KB of high-speed cache memory, a 4-KB translation lookaside buffer and a floating-point coprocessor).

As a result of this advanced technology, the HP 3000 Series 950 provides seven MIPS of processing performance in multiuser applications, an unprecedented amount of performance for any VLSI chipset in the industry. Series 950 is the highest-performance HP 3000 system, offering performance comparable to the DEC VAX 8550 and 8700 and IBM 4381 systems.

Series 950 base system includes 32 MB of main memory, MPE XL operating system, relational and network database management systems, and HP's System Dictionary.

Since Series 950 uses fewer parts to achieve high performance, it also provides high reliability, resulting in low hardware-support prices and low cost of ownership.

Series 930, with 1 Mbit RAM technology, now includes 32 MB of main memory, which can be expanded to 96 MB. Series 950, which also uses 1 Mbit RAM technology, can support main memory configurations up to 128 MB.

Chance also said customer migration experiences to date confirm that the Series 930 and 950 are compatible with the rest of the HP 3000 family.

HP also announced enhancements to its current line of HP 3000 systems, including faster system backup and increased system security.

## Wollongong, HP Sign Second Development Agreement

*Wollongong To Develop WIN/H3000 For 3000s Running MPE/V*

The Wollongong Group (Palo Alto, CA) has signed its second major development agreement with Hewlett- Packard in the last nine months.

The new agreement calls for Wollongong to develop WIN/H3000, a communications software product for HP 3000 computers running HP's MPE/V operating system. Based on the U.S. Defense Department's standard Defense Data Network (DDN) communications protocols, WIN/H3000 will give HP 3000 users the ability to access non-HP 3000 systems on the Defense Data Network for the first time.

The product is scheduled to be released during the second quarter of 1988.

**Version 3.0 Seems Far Away, But HP Support Keeps It In Focus.**

**INDUSTRY WATCH**

**Lonni Wright**

# MAP: Closer To Reality?

As users wait for version 3.0 of the Manufacturing Automation Protocol (MAP), the specifications of which were released several weeks ago, they focus their attention on other matters.

Experts say interest in MAP probably won't increase until June 1988 when products based on the new version of 3.0 of the networking specifications are released. The MAP/TOP (Technical and Office Protocol) Users Group, which backs the emerging factory networking standard, will host the Enterprise Event 88 International that month, where the first public demonstration of MAP 3.0 is expected.

Many companies *are* testing their ideas on MAP 2.1, however, because the migration to 3.0 is becoming more clearly defined.

BUT ADDING TO THE DISINTEREST in supporting MAP is the press it has received on a controversy between Digital Equipment Corporation and the MAP/TOP Users Group.

The controversy began a few months ago when DEC Chairman Ken Olsen made several public statements that implied there was no need for a new factory networking standard.

Olsen said Ethernet, on which the company's DECnet networking system is based, is sufficient for the factory environment and a workable alternative to the current MAP-specific physical media, the IEEE 802.4 broadband cable.

To make matters worse, at a spring meeting of the MAP/TOP Users Group held in Pittsburgh, a DEC vice president tried to clarify issues surrounding DEC and MAP and instead implied that the users group was in favor of adopting other physical media specifications.

The added controversy came about when the DEC official announced an agreement between DEC and General Motors Corp., the initial proponent of MAP, under which the two companies will reopen a process to test and evaluate other physical media for use in a MAP network.

*. . . 802.4 technology is the proper media . . .*

GM officials defended the company's belief that 802.4 technology is the proper media for the factory environment and that it agreed only to evaluate new data on Ethernet. Many supporters believe that the testing only will prove again that 802.4 is a better choice in most CIM installations.

HEWLETT-PACKARD, A FOUNDING member of the MAP Users Group, is enthusiastic about MAP and remains at the forefront of efforts to establish it as the international standard for multivendor factory networks.

The company actively is supporting user efforts to use MAP 2.1 now in technical experiments and pilot projects and to develop their internal expertise on MAP. HP has been working on the MAP specification since its inception in 1980 and is committed to developing standard products to implement it.

HP announced its first standard MAP products in September 1986. It premiered its MAP 2.1 interface products for HP 1000 workcell control systems and HP 3000 systems at a MAP Users Group meeting in Ann Arbor, Michigan.

HP has begun work on the Ford Motor Company's first MAP pilot, which is projected for start-up late this summer at the Electrical and Electronics Division Plant (Rawsonville, MI).

A 10-MB MAP 2.1 communications network will be used to integrate computers with automated test systems and programmable controllers interfaced to robots, machining equipment, conveyors and inspection gauges.

HP is supplying HP 1000 computer systems, a host computer and a MAP gateway to interface with programmable logic controllers.

The company also has a development program under way for its new Precision Architecture computers and has signed a joint marketing agreement with Ungerman-Bass (Santa Clara, CA), a supplier of network products.

The agreement calls for joint design, installation and maintenance of broadband local area cable systems and terminal connection networks.

A letter of intent to Industrial Networks, Inc. (INI, Santa Clara, CA) from HP is the first step in developing an agreement between the two companies to jointly market MAP products.

The agreement enables both companies to provide complete factory networking solutions to HP customers.

HP's other MAP accomplishments include the successful completion of the first operational MAP pilot at GM. In August 1985, a MAP Revision 2.0 pilot was completed at a GM plant in Marion, Indiana, becoming the first functional MAP installation within General Motors.


TurboIMAGE
DB GENERAL
Join the party!
BRADMARK
ENTER 164 ON READER CARD

HP was chosen for a major role in the GM Truck and Bus Program, the most significant MAP installation at GM for 1986. Major vendors participating in the contract include HP with up to 25 workcell controllers per plant, DEC with area management computers and INI with networking products.

Many of HP's own manufacturing divisions plan to utilize MAP. The Roseville Networks Division (Roseville, CA), the company's pilot plant, is also the base for a majority of HP's MAP product development programs.

At Roseville, HP has installed broadband cabling throughout the plant. Point-to-point computers and terminal connections are being installed and migration to a full MAP network will take place when standard products become available.

In addition to its MAP support, HP supports TOP, another emerging international communications protocol which is intended to provide multivendor networking for office and engineering environments.

THE COMPANY HAS long been involved in the development of international standards. As a founder of the Corporation for Open Systems, HP hopes to advance the adoption of international interconnect standards, such as the Open Systems Interconnect Standard, throughout the industry and to provide a consistent set of test methods and certification procedures. It also spearheaded the worldwide adoption of IEEE Standard 488, which enabled instruments and computers to communicate with each other.

Considering the support HP gives MAP, the present disinterest in moving to the standard most likely is just the quiet before the storm of MAP 3.0 products to be released next year. With those developments, the industry will be closer to making MAP a reality. Then manufacturers will be forced to choose between HP and other CIM vendors only on the basis of the quality of their CIM products, not which company has the best proprietary network. ■

# A disk drive that three can enjoy.

## Hewlett-Packard users, drink up.

If the high cost of storage and networking has left you stranded, you're in for a treat.

Introducing MultiPort. The HP-compatible disk drive that can be shared by either two or three computer users.


MultiPort is one complete, cost-effective system for both storage and backup. That's because each user is allotted one *equal* amount of storage on the fixed disk. In addition, they also have access to a 20MB removable Winchester cartridge, and a Shared Information Space (SIS) that serves as a networking alternative.

So it's like having three (or two) separate hard disks and a backup unit on one space-saving system.

## Share and share alike.

If you want to share files from one user to another, you just download your data onto the removable cartridge (just like you would a floppy), remove the cartridge, pop it into your drive and away you go.

There's no problem of accidental data address. And because the 20MB cartridges are removable, you have unlimited storage capacity. You have unlimited security as well, since you can lock the cartridges up.

## SIS–The refreshing difference.

MultiPort's unique Shared Information Space is also available for low-cost networking. When the SIS is configured, it acts like an extra, "separate disk" that everyone can share.

Thus, everyone can access it and work off the same file, but only one at a time. Security and traffic features prevent simultaneous access of the SIS "disk."

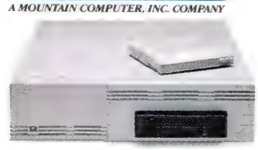
## Storage choices that satisfy.

MultiPort is available in 20, 40, 50 or 70MB models. Software is included. The system is compatible with HP9000 Series 200/300/500 computers and fits HP's ITF footprint specifications for easy integration into your HP environment. Choices include:

8220: 20MB fixed & 20MB removable
8240: 40MB fixed & 20MB removable
8250: 50MB fixed & 20MB removable
8270: 70MB fixed & 20MB removable

For more information on all our MultiPort systems, call or write today. Bering Industries, Inc., 280 Technology Circle, Scotts Valley, CA 95066, 408-438-8779. Inside California, call **800 533-DISK.** Outside California, call **800 BERING 1.**

**BERING**™
A MOUNTAIN COMPUTER, INC. COMPANY



**Innovative Storage for Hewlett-Packard.**

Footnotes: MultiPort and Shared Information Space (SIS) are trademarks of Bering Industries, Inc. HP9000 Series 200/300/500 are trademarks of Hewlett-Packard. © Bering Industries, Inc.

ENTER 196 ON READER CARD

## HP BASIC Language Processor

Hewlett-Packard recently announced its premiere instrument-control language for the HP Vectra PC.

The HP BASIC Language Processor consists of an HP Vectra PC plug-in card with HP-IB interface, HP BASIC 5.0 (Rocky Mountain) and support software. This is the same BASIC that runs on HP 9000 Series 200/300 computers, supplied on 5¼-inch, DOS format disks.

Now, users who are required to use personal computers do not need to forego the time-saving advantages of HP BASIC or rewrite their HP 9000 Series 200/300 programs. When installed in an HP Vectra PC, the new subsystem delivers HP's powerful controller environment and access to PC-DOS application software.

Other features include access to industry-standard LAN (SRM and IEEE 802.3 available), computer and I/O performance greater than HP 9816, common DOS/BASIC fail structure that allows data analysis via popular PC software, and foreground/background operation.

HP's BASIC 5.0 aids program development through interactive editing, powerful syntax checking and the ability to search and replace strings or blocks of lines.

Advanced features such as structured programming with PASCAL-like constructs (such as if-then-else blocks) improve programmer productivity and simplify on-going support. Independent subprograms can be used throughout the code, while local variables, global storage and runtime linking are available for advanced BASIC programmers.

Advanced I/O simplifies many complex tasks such as interrupts, data transfer, overhead processing, printing, mass storage and plotting, all of which are critical to intelligent instrument control. HP BASIC also features the computational power found in FORTRAN, ALGOL and other highly mathematical languages.

The language processor board, which includes a Motorola 68000 CPU and up to 4 MB of RAM, emulates an HP 9000 Series 200 workstation. It includes built-in HP-IB and standard HP 9000 Series 200/300 DIO interface. BASIC is run like any other PC-DOS application and is compatible with DOS files and peripherals.

Since HP BASIC runs on its own processor board, not on the HP Vectra PC's Intel 80286, simultaneous operation of BASIC programs and PC-DOS applications is possible.

With optional hardware, the processor operates on HP networks such as SRM and OfficeShare, and can pass files to ARPA (IEEE 802.3) networks as well.

The HP BASIC Language Processor includes:

■ *MC 68000 processor card (523 KB of RAM expandable to 4 MB, built-in HP-IB interface, DIO connectors, connectors for ROM expansion board).* ■ *HP BASIC 5.0 software.* ■ *PC Emulator software.* ■ *Manuals and keyboard overlays.*

Accessory products include:

■ *RAM expansion board and 512-KB expansion kits.* ■ *GPIO interface.* ■ *Shared Resource Manager (SRM) interface.*

**Enter 900 on reader card**

## HP Announces Lotus Driver 1002

Hewlett-Packard's new Lotus Driver 1002, a three-disk package of Lotus Development Corp.-compatible peripherals software drivers, enables all versions of Lotus's 1-2-3 and Symphony to run on HP's personal printers and plotters.

Printers and plotters supported include the LaserJet printers, ThinkJet, QuietJet Plus, the 2930 series printers, the Colorpro and the 7470, 7475 and 7550 plotters, operating both in paper and transparency modes.

The package is available for $25 per disk.

**Enter 902 on reader card**

## Infotek To Introduce 5.0 BASIC Compiler

Infotek Systems will introduce a compiler for HP's newest operating system, BASIC 5.0.

The new compiler, the BC305, will be bundled with Infotek's MB305 fast matrix binary, which performs MAT statements two to three times faster than the HP MAT binary.

It will be compatible with all HP series 200/300 computers, including the new 330 and 350, and will support all BASIC 5.0 statements, including complex arithmetic.

The BC305 also will provide unsecured execution support allowing users to run compiled code on multiple systems.

Contact Infotek Systems, 1045 S. East Street, P.O. Box 65008, Anaheim, CA 92805-8508; (714) 956-9300.

**Enter 906 on reader card**

## Lower Prices On Vectra CAD/CAE

Effective April 1, HP lowered the list price of its 82964E HP Vectra PC CAD/CAE System.

The 82964E CAD/CAE System is a preassembled, high-performance bundle ideal for running computer-aided design and other "power-user" applications. It includes:

■ *640-KB HP Vectra PC SPU with 1.2-MB floppy drive.* ■ *IBM PC/AT compatible, 40-MB hard disk subsystem.* ■ *Enhanced color graphics (EGA) display system with tilt/swivel base.* ■ *High-speed (8 MHz) 80287 math coprocessor.* ■ *Serial/parallel card.* ■ *HP Vectra PC DOS/PAM*

**Enter 901 on reader card**

## MDSS Package Enhanced

MDSS, Inc., developers of software solutions for manufacturers, has enhanced its software package, MDSS. The addition provides organizations the ability to run multiple companies from a single copy of MDSS as well as to provide training.

Each division or company now can configure MDSS to its specific need while using the same source code. For example, one corporate entity can make custom changes to its database while another entity can run on the standard MDSS package.

Contact Manufacturing Decision Support Systems, Inc., 300 East Ohio Building, 1717 East 9th Street, Cleveland, OH 44114; (216) 861-8100.

**Enter 905 on reader card**

***RS/1 data analysis software now is available on HP 9000 Series 300 workstations.***

## BBN Ports RS/1 To 9000 Series 300

BBN Software Products Corporation has announced a new version of its RS/1 software package for HP 9000 Series 300 engineering workstations under the HP-UX operating system. RS/1 software is a data management and analysis system used in engineering, scientific and manufacturing environments.

RS/1 software's capabilities include data management and analysis, statistics, curve fitting, modeling and graphics.

RS/1's programming language, RPL, is used in industry to develop customized systems for laboratory and factory automation. RS/1 also operates on HP's Vectra; DEC's VAX, MICROVAX and PDP computers; IBM's PC AT, PC XT, 3270 PC and RT PC computers; and IBM's 9370, 4300 and 3000 Series mainframes.

Prices range from $3,900 per single user copy on Series 310 workstations to $9,900 on multiuser Series 350 systems.

Contact BBN Software Products, 10 Fawcett Street, Cambridge, MA 02238; (617) 864-1780.

**Enter 903 on reader card**

## CIRCA Announces Port To Spectrum

CIRCA Management Systems, Inc., has completed a port of REFLECTS to the HP Spectrum 9000/840 precision architecture system. Simultaneously, the company announced its intention to market the product directly to Selected Software Suppliers and to selected VARs.

REFLECTS permits commercial applications written in Interpretive Business BASIC languages to be translated into standard Kernighan & Ritchie C code. The product comes complete with a very broad run-time library preconfigured for use with Informix C-Isam, and adaptable to Informix, Oracle, C-English and other 4GL databases.

CIRCA Management Systems, Inc., is located at 17610-21 Beach Blvd, Huntington Beach, CA 92647; (714) 841- 6038.

**Enter 904 on reader card**

## File/Swap PC V2.0 Now Available

A Gentle Wind Inc. has released File/Swap-PC Version 2.0. New features in this HP to Microsoft file translation software include automatic adaptation to both the HP logical format and the physical medium.

HP physical disk parameters may be cached without user intervention, or set with a one-touch menu prior to disk formatting.

An improved copy algorithm speeds the copy process. A complete 270-KB HP disk with 60 file entries can be copied into MS-DOS format on an HP Vectra Model 65 in just over five minutes.

File/Swap-PC V2.0, priced at $249.00, now offers complete MS-DOS hard disk subdirectory support with easy one-touch commands. HP-CP/M disks can be "wildcard" selected for copying with the same naming standards as MS-DOS 3.0.

Contact A Gentle Wind Inc., Box 3103, Albany, NY 12203.

**Enter 917 on reader card**

## NEK Cables Compatible With IEEE 802.3

Cables manufactured by NEK now are compatible with the IEEE 802.3 networking standard.

Included are 50 ohm trunk coax, four-pair transceiver and Thinnet coax cable. All cables are made with FEB dielectrics for high-speed data transmission.

NEK Cable, Inc., manufactures a line of plenum fluorocarbon insulated cable and assemblies for baseband and broadband local area computer networks, data transmission, telecommunication systems, energy management and alarm systems for plenum use without conduit.

NEK Cable, Inc., is located at 2150 Fifth Avenue, Ronkonkoma, NY 11779.

**Enter 913 on reader card**

## Okidata Lowers Printer Price

Okidata's Laserline 6 printer is now available for $1,795.

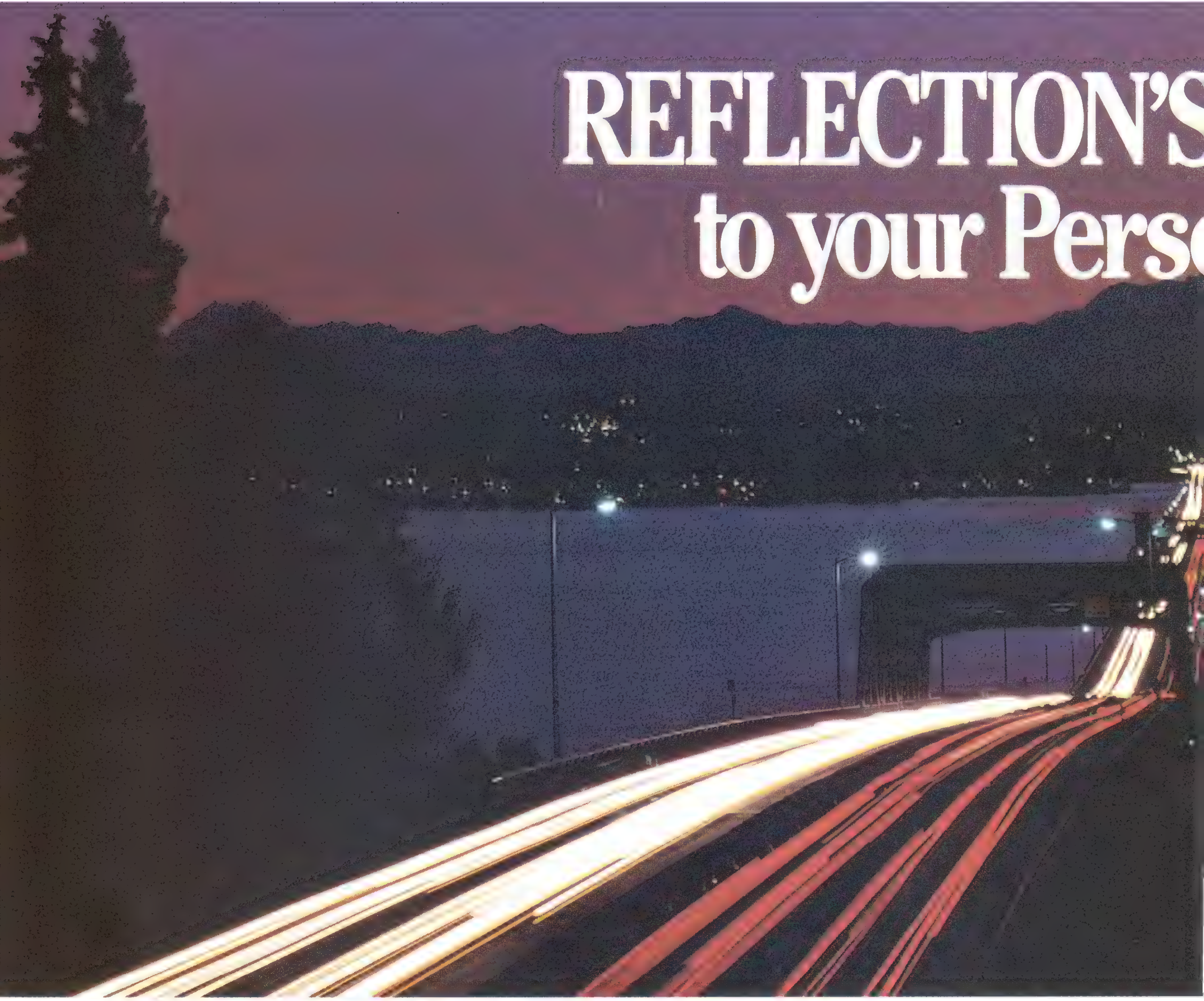
The six-page-per-minute printer and advanced interface emulates HP's LaserJet Plus and includes 15 resident fonts.

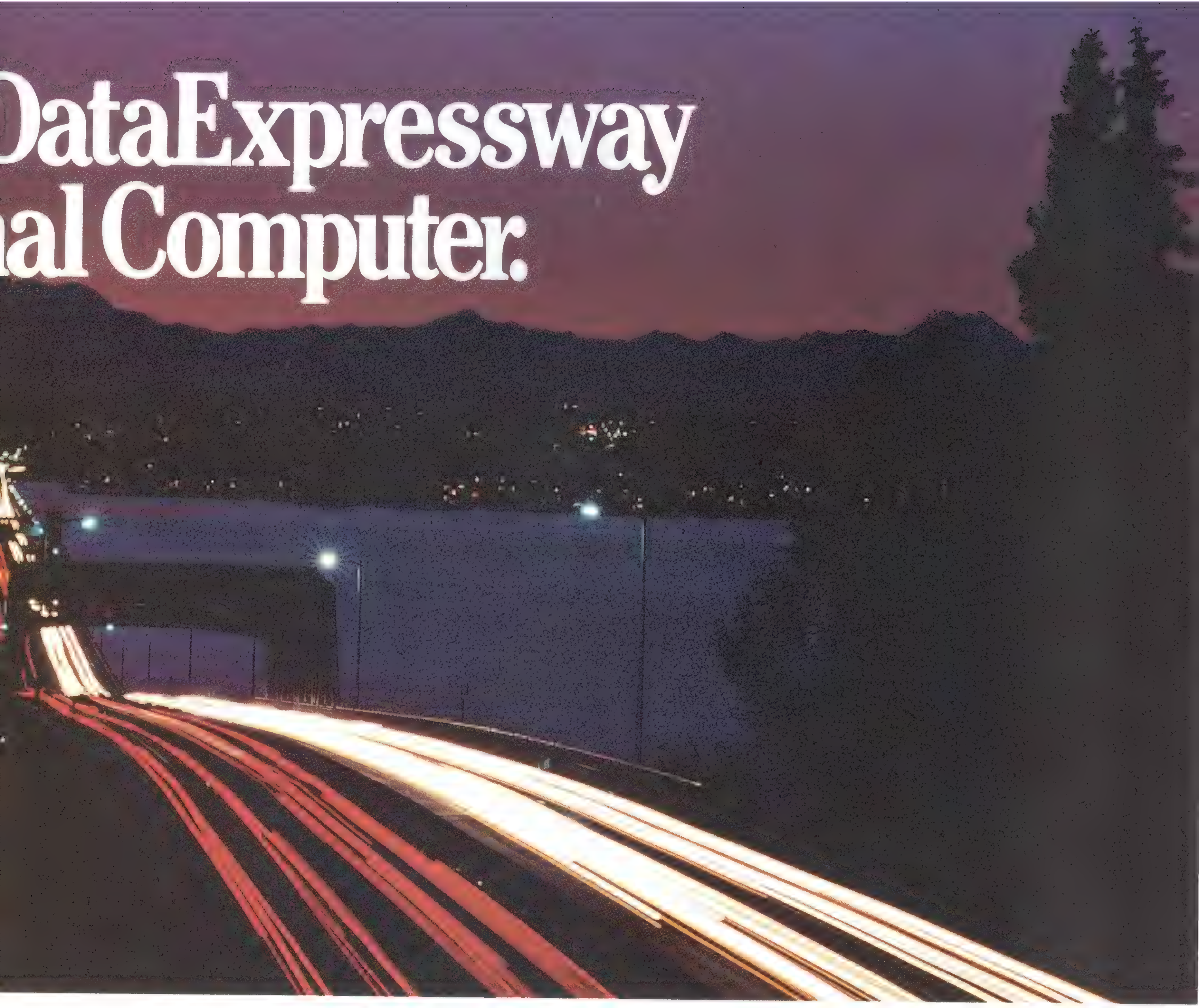
In addition, Okidata has announced an interface that allows up to three users to share the Laserline 6, making it a suitable replacement for daisywheel printers in general office applications.

Contact Okidata at 532 Fellowship Road, Mount Laurel, NJ 08054; (609) 235-2600.

**Enter 912 on reader card**


***Okidata's Laserline 6.***


REFLECTION'S
to your Pers




**DataExpress is widely accepted as the most comprehensive application and data integration facility for the HP 3000 and PCs linked by Reflection.**

This user-oriented system extracts data from Image, TurboImage, KSAM, MPE files, Omnidex and Relate/3000. The data can then be manipulated, computed, redefined and reformatted for input to most major spreadsheet and database programs which run on IBM or Macintosh personal computers.

Designed for use by non-DP personnel, DataExpress allows a system administrator to control the capabilities of each individual user. The result is a simple yet sophisticated menu that guides you through the steps of data access, selection, computation, and reformatting.

ReportExpress – a companion product – provides end users with *ad hoc* and fully formatted reporting capabilities.

Together, Data Express and Reflection allow for easy, yet comprehensive information exchange from your HP 3000 to your favorite PC program.

Call us at **213-823-4577** or outside California **1-800-445-9519**.

DATAEXPRESS™

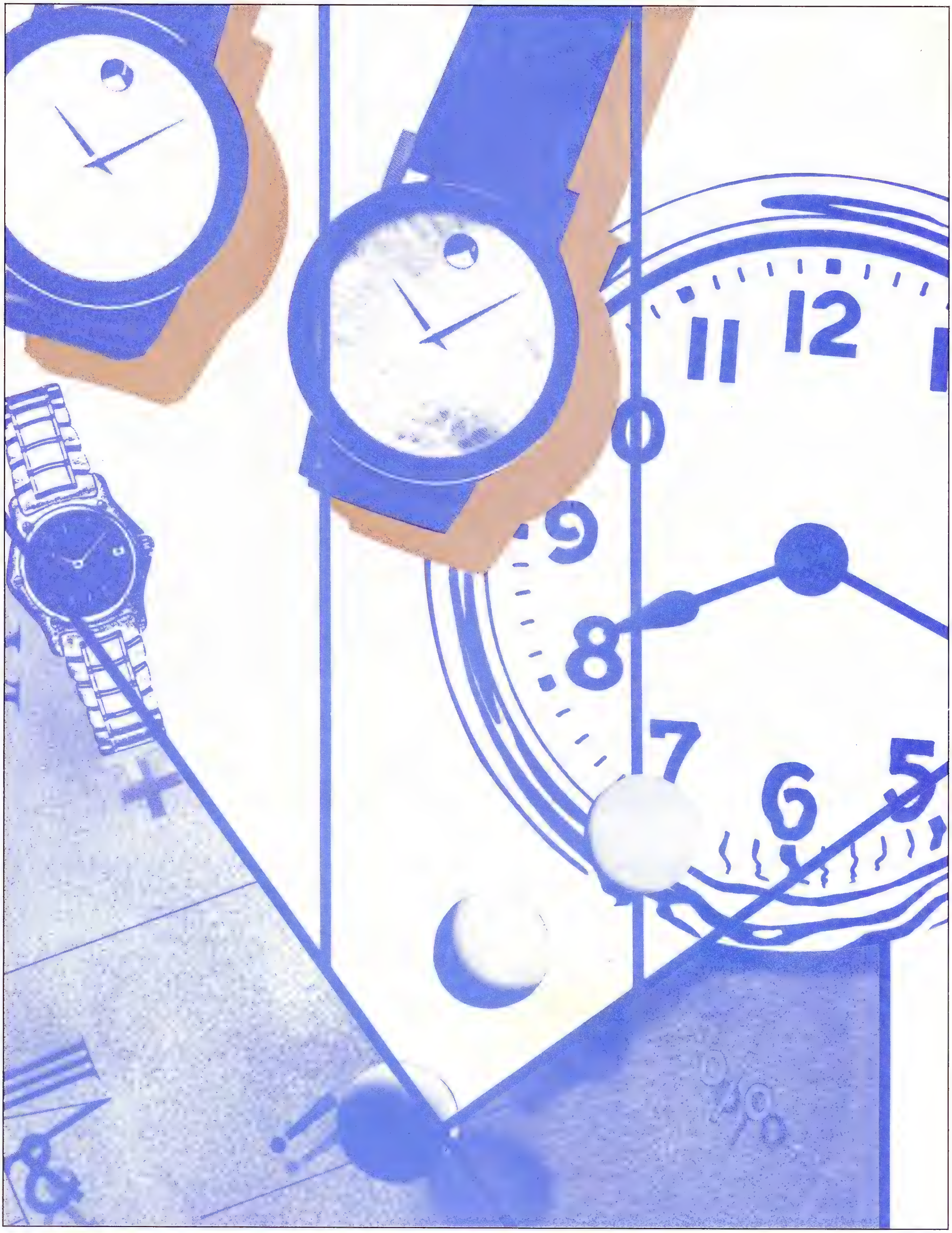
FROM

IMACS

IMACS SYSTEMS CORPORATION

4676 Admiralty Way, Suite 507, Marina del Rey, CA 90292
LOS ANGELES · ROTTERDAM · LONDON · CALGARY

ENTER 105 ON READER CARD



*By Adding A Little New Functionality And A Lot Of Performance-Tuning, UNIX Systems Can Support More Demanding Real-Time Applications Like Those Found On The Factory Floor.*

# UNIX For Real-Time Markets

By Suzanne M. Doughty, Sol F. Kavy, Steven R. Kusmer, Douglas V. Larson

UNIX is the operating system for scientific supercomputers and PCs, and is on the desks of software professionals and CEOs. However, one of the final frontiers for the acceptance of UNIX systems is in the real-time marketplace, and for good reason: The real-time customer is *the* most demanding customer. His demands fall within three categories:

**■ Performance**

Real-time applications are measured primarily by their performance. Therefore, real-time customers will expect to squeeze the last ounce of performance out of a real-time system to meet their needs, and they sometimes will take measurements their computer vendor never expected.

The performance characteristics they measure typically are in terms of response time or throughput. An example of a response time measure is "How long after the receipt of an interrupt from my parallel I/O card can the system run my process which was waiting for that interrupt?"

An example of a real-time throughput measure is "How long will it take for me to push my two gigabytes of data from my device to the file system?"

The real challenge is that both questions often will be asked by the same customer.

**■ Determinism**

Customers expect a real-time system to react in a deterministic manner. For example, it's not enough to have good response most of the time — you must provide good response *all* of the time. Real-time customers often build a computer into a system that has unforgiving constraints, usually because the system is controlling or monitoring other devices or machinery.

As an example, a real-time computer built into a steel mill whose steel travels at 30 mph will be expected to respond quickly to an alarm condition. If the computer unexpectedly becomes busy for a whole second, the steel in the mill will have traveled 44 feet and possibly could have been strewn all over the mill floor.

> Customers expect a real-time system to react in a deterministic manner.

**■ Flexibility**

In the end, it's real-time customers who truly know best how a real-time computer can solve their applications needs. Customers must be provided with tools for writing their own drivers and measuring system performance. They must be provided with source code, because they choose to understand in-depth how a system performs and they might want to tune it for their application.

On the other hand, vendors of real-time computers must be humble, because real-time customers are glad to tell them how to build their systems.

Here we will present a definition of a real-time system and explain the real-time features implemented on the HP 9000 Series 800 Model 840, the first of HP's new Precision Architecture computer line, running HP-UX.

A REAL-TIME SYSTEM IS one that can respond in a deterministic and timely manner to events in the real world. This could mean either large amounts of data that must be processed fast enough to prevent losing the data (data throughput), or discrete events that must be recognized and responded to within certain time constraints (response time).

Specific time requirements depend on the real-time application. For example, in an airline reservation system, a customer calls an airline representative and requests a reservation on a certain flight. The customer waits on the phone while the real-time system processes his transaction and then responds to the request either with a confirmation or a "flight is already booked" message.

The response time requirement for this application is on the order of a second or so, while the customer waits on the phone. If the system fails to meet this requirement, we're left with frustrated customers and perhaps loss of business.

An example of a more demanding real-time application is process-monitoring and control in a steam-powered electric plant. Sensors are used to measure variables such as pressure and level of water in the boilers, and speeds of turbines and generators. For example, if the boiler pressure gets too high, a real-time system must respond immediately with the appropriate action (either reducing the heat source or initiating some cooling action).

The response time requirement for this application is on the order of tens of milliseconds. If the system fails to meet this requirement, we could be left with incorrect electrical output or perhaps extensive damage caused by an exploded boiler.

The above applications are just two of the many and varied examples of real-time applications. *Figure 1* presents and categorizes additional real-time applications. It's important to note that this list is by no means comprehensive. Its purpose is to show the variety and pervasiveness of real-time applications.

Given a definition of real time and some sample real-time applications, the next question is "How can the UNIX operating system be augmented to meet the requirements of real-time applications?"

While using System V as a base, HP-UX answers this question in two parts:

■ *By incorporating functionality from 4.2 BSD and adding new functionality from HP.*

■ *By doing performance-tuning on the kernel and file system.*

To better understand this approach, it's helpful to be familiar with HP's goals for adding real-time capability to the UNIX operating system:

■ *Any real-time features implemented must not prevent System V Interface Definition (SVID), Issue 1, compatibility.*

■ *Wherever possible, real-time features should be adopted from either System V or 4.2 BSD. Only where a needed real-time feature does not exist should HP add a new feature.*

■ *Real-time features must be portable.*

■ *Performance-tuning must be transparent to user processes.*

■ *Real-time response must be comparable to real-time response on the HP 1000 A900 (HP's top-of-the-line, real-time A Series computer).*

HP-UX on the Model 840 has met these goals. In addition, HP is lobbying through standards-setting bodies to encourage their adoption of HP-UX's real-time features as part of an existing or evolving standard such as SVID or IEEE P1003. HP-UX on the HP Series 300 and 800 has passed the System V Verification Suite (SVVS).

FIGURE I

| General Real-Time Applications | Examples |
|---|---|
| process monitoring and control | petroleum refinery<br>paper mill<br>chocolate factory |
| data acquisition | pipe-line sampling<br>data inputs from a chemical reaction |
| communications | monitoring and controlling satellites<br>telephone switching systems |
| transaction-oriented processing | airlines reservation systems<br>online banking (automatic tellers)<br>stock quotations systems |
| flight simulation and control | autopilot<br>shuttle mission simulator |
| factory automation, factory floor control | material tracking, parts production<br>electronic assembly<br>machine or instrument control |
| transportation | traffic light systems<br>air traffic control |
| interactive graphics | image processing<br>video games<br>solids modeling |
| detection systems | radar systems<br>burglar alarm systems |

***General real-time applications and some examples.***

NOW LET'S LOOK at the real-time features of HP-UX on the Model 840, their origin (either System V, 4.2 BSD or HP) and how each one addresses certain concerns about the real-time capability of UNIX systems.

The following features provide real-time capability to HP-UX:

**Added Functionality**

- *Priority-based preemptive scheduling*
- *Process memory-locking*
- *Privilege mechanism to control access to real-time priorities and memory-locking*
- *Fine timer resolution and time-scheduling capabilities*
- *Interprocess communication and synchronization*
- *Reliable signals*
- *Shared memory for high-bandwidth communication*
- *Asynchronous I/O for increased throughput*
- *Synchronous I/O for increased reliability*
- *Preallocation of disk space*
- *Powerfail recovery for increased reliability*

**Performance-Tuning**

- *Kernel preemption for fast, deterministic response time*
- *Fast file system I/O*
- *Miscellaneous performance improvements*

PRIORITY-BASED PREEMPTIVE SCHEDULING lets the most important process execute first, so it can respond to events as soon as possible. The most important process executes until it sleeps voluntarily or finishes executing, or until a more important process preempts it.

Priority-based means that a more important process can be assigned a priority higher than other processes, so the important (high-priority) process will be executed before others.

Preemptive means that the high-priority process can in-

terrupt or preempt the execution of a lower priority process, instead of waiting for it to be preempted by the operating system when its time slice is completed or it needs to block.

The scheduling policy of traditional UNIX systems strives for fairness to all users and acceptable response time for terminal users. The kernel dynamically adjusts process priorities, favoring interactive processes with light CPU use at the expense of those using the CPU heavily.

Users are given some control of priorities with the **nice(2)** system call, but the nice value is only one factor in the scheduling formula. As a result, it's difficult or impossible to guarantee that one process has a priority greater than another.

Therefore, each process in a traditional UNIX system effectively has to wait its turn, no matter how important it might be to the real-time application.

HP-UX presents a solution to this problem by adding a new range called real-time priorities. Priorities in this range do not fluctuate like priorities in the normal range. Any process with a priority in the real-time range is favored over any in the normal range, including those making system calls and even system processes.

Important as real-time processes are, interrupt processing is given priority over them. If several real-time processes have the same priority, they are time-sliced.

Processes with real-time priorities are favored not only for receiving CPU time, but also for swapping and file system accesses. Real-time processes are the last to be swapped out (except for locked processes — see below), and the first to be swapped in. File system requests for real-time processes go to the head of the disk request queue.

All of this preferential treatment gives real-time processes very good response, but at the expense of the rest of the system. (There's no free lunch!)

Real-time priorities are set by the user either programmatically with HP's new **rtprio(2)** system call, or interactively with the **rtprio(1)** command. By default, processes are time-shared and continue to be executed according to the normal scheduling policy.

Aside from setting a process to a real-time priority, the **rtprio(2)** system call and **rtprio(1)** command can be used to read the priority of a real-time process and change it to be time-shared.

ANOTHER IMPORTANT FEATURE in a real-time system is the ability to lock a process in memory so it can execute without waiting to be paged in or swapped in from disk. In the UNIX and HP-UX operating systems, processes normally aren't locked in memory; they're swapped and/or paged in from disk as needed.

The time required to swap in a process, or page in one or more pages of a process, can range from several milliseconds to several seconds, which violates the response time requirements of many real-time applications.

HP-UX has adopted a solution to this problem from System V. The **plock(2)** system call allows a process to lock its executable code and/or its data in memory to avoid unexpected swapping and paging. Also, a process can lock additional data or stack space with the **datalock(3C)** subroutine, and lock shared memory segments as needed with the **shmctl(2)** system call.

BECAUSE THE PRIORITY-SCHEDULING and memory-locking features of HP-UX are quite powerful, it's desirable to allow only certain users to access them.

If, for example, all users had access to these capabilities, they potentially could set all of their processes to a high real-time priority and try locking them in memory, which would defeat the purpose of the real-time system.

Or, a novice user could assign a real-time priority of 0 to a process that happens to execute in an infinite loop, thus locking up the entire system.

To prevent scenarios like these, HP-UX created a feature called privilege groups, which enable certain users (other than just the superuser) to access the powerful real-time priority and memory-locking features of HP-UX. A privilege group is a group to which the superuser assigns privileges. Existing privileges are real-time priority assignment (RTPRIO), memory-locking (MLOCK) and a third privilege which isn't related to real-time functionality.

The superuser assigns one or more of these privileges to one or more groups with HP's **setprivgrp(2)** system call or **setprivgrp(1)** command, and assigns certain users to become members of these groups with the 4.2 BSD-based **setgroups(2)** system call. You can retrieve the groups you belong to with 4.2 BSD's **getgroups(2)** system call or **groups(1)** command, and retrieve the privilege groups you belong to with the **getprivgrp(2)** system call or **getprivgrp(1)** command.

ANOTHER IMPORTANT FEATURE in a real-time operating system is fine timer resolution and time-scheduling capabilities. For high-resolution clock-based applications, both repetitive and nonrepetitive, it's important to be able to execute a process or subroutine at a precise time. For example, a real-time application might require various sensor readings at 20 millisecond intervals.

Standard features in System V that deal with time, such as **alarm(2)**, which has a resolution of one second, and **crontab(1)** and **at(1)**, which have a resolution of one minute, aren't precise enough for many real-time applications. Therefore, HP-UX has adopted a solution from 4.2 BSD known as interval timers.

Each process can enable its own interval timer to interrupt itself once or at repeated intervals, with whatever precision the underlying hardware and operating system can support.

The interval is defined in units of seconds and micro-

Finally—PC-based publishing that fits not just one desktop, but your whole corporation.

Now you can give your users a complete—and completely supported—MS®-DOS-compatible desktop publishing solution from Hewlett-Packard.

So they can have the high quality and cost savings they want. And you can have a system that fits in with your long-term office automation strategy.

The Vectra Publisher PC—HP's PC/AT-compatible Vectra personal computer bundled with the largest-selling page composition software, PageMaker® from Aldus—is the foundation. HP's industry-leading LaserJet printer and ScanJet scanner complete the solution.

And because HP has made them an integral part of our office solutions, desktop publishing systems connect to your HP 3000, linking into information resources throughout your company.

And Hewlett-Packard offers the nationwide—or even worldwide—service and support your company-wide desktop publishing solution will require.

You see, we've looked beyond the insular needs of isolated desktop publishers. We've taken the global view of your entire business.

So why buy a "communications tool" that may not speak your language? Hewlett-Packard desktop publishing solutions are ready for delivery now. Call 1-800-367-4772, Dept. 234A, for your free desktop publishing information kit—and find out about our special, limited-time, money-saving offer.

MS®-DOS is a U.S. registered trademark of Microsoft, Inc
PageMaker® is a U.S. registered trademark of Aldus Corporation

**HEWLETT PACKARD**

# Hewlett-Packard desktop publishing.

# Because no desk should be an island.

ENTER 123 ON READER CARD

seconds to keep the timer interface portable despite the system-dependent resolution. For HP-UX on the Model 840, the system clock resolution is 10 milliseconds.

A REAL-TIME OPERATING SYSTEM must provide interprocess communication and synchronization facilities. These are important, because real-time applications often involve several asynchronous processes that need to exchange information.

For example, in a manufacturing environment there might be several dedicated computers on a production line, where each executes a process that controls the movements of parts and actions on those parts (such as soldering, molding or welding). A supervisory process that runs on a more powerful computer might monitor the activities of the controller processes. If some type of alarm condition occurs on the production line, the supervisory process can initiate a slowdown or shutdown action.

A second process that runs on the supervisory computer could keep track of the inventory levels of each part and inform a third process when more parts must be retrieved and sent to a particular dedicated controller.

As you can see, the processes in the supervisory computer must communicate with each other and with the dedicated processes on the production line computers. This is just one example of a group of processes involved in a real-time application that must communicate with each other to get the job done.

Pipes and signals are common interprocess communication facilities in UNIX. A pipe is essentially an I/O channel through which data is passed with the **read(2)** and **write(2)** system calls. An advantage of using a pipe is that it provides synchronization by blocking reader processes when the channel is empty and blocking writer processes when the channel is full.

The disadvantages of using pipes are:

■ *They require the communicating processes to have a common ancestor process that sets up the channel.*

■ *They often are slow, because the kernel has to copy the data from the writer process to the system buffer cache and then back again to the address space of the reader process.*

Many UNIX systems including HP-UX support named pipes, which overcome the first problem but still have the performance penalty of copying the data.

A signal is essentially a software interrupt sent to a process by the kernel or by a user process. A process can install a handler for almost any signal, and the handler will be executed when the signal is received. Signals can be a good event or alarm mechanism, because one process can send a signal to inform another process that an event occurred, and then the other process immediately can enter its handler to respond to the event.

The disadvantage of using signals are:

■ *They pass little or no data (not even who the sender process is).*

■ *They traditionally are unreliable when sent repeatedly or when a process tries to wait for a signal.*

[ Shared memory allows the highest communication bandwidth . . . ]

HP-UX, therefore, has adopted a reliable signal interface from 4.2 BSD, in addition to the System V signal interface. The 4.2 BSD signal interface solves the reliability problems of the System V interface, but it's more complicated to use. HP-UX has modified its signal interface to emulate completely both the standard System V interface and the 4.2 BSD interface.

HP-UX also has adopted three IPC facilities from System V: shared memory, semaphores and messages. These facilities allow communication and synchronization among arbitrary unrelated processes.

An elaborate semaphore facility allows solutions to both simple and complex synchronization problems. A message-passing facility allows transfer of data, along with the ability to prioritize messages.

The most important IPC facility for real-time applications is shared memory. Two or more processes can attach the same segment of memory to their data space and then write to and read from it. Shared memory allows the highest communication bandwidth, since data does not have to be copied to be communicated.

Recall that a shared memory segment can be locked in memory to provide optimal performance for the communicating processes.

ASYNCHRONOUS I/O OVERLAPS WITH process execution or other I/O, typically resulting in increased throughput. Both UNIX and HP-UX implement system asynchronous I/O to certain drivers, but HP-UX allows you to communicate with some drivers that do system asynchronous I/O, so you can take advantage of their asynchronous abilities.

System asynchronous I/O occurs when the system does asynchronous I/O for a process while the process continues to execute. Two examples where the UNIX and HP-UX operating systems do system asynchronous I/O for user processes are writing to the file system and reading from a terminal when there are enough characters already in the terminal buffer to satisfy the **read(2)**.

HP-UX implements system asynchronous facilities for terminals, pipes, named pipes and sockets. The system asynchronous I/O facilities that HP-UX provides for terminals are:

■ *The nonblocking I/O facility: Before launching an I/O request, a*

FIGURE 2

| Real-Time Function | Associated System Call | Origin |
|---|---|---|
| Priority-based preemptive scheduling | rtprio(2) | HP |
| Process memory-locking | plock(2) | System V |
| Privilege groups | getprivgrp(2)<br>setprivgrp(2) | HP and 4.2 BSD |
| Fine timer resolution and time-scheduling capabilities | setitimer(2)<br>gettimeofday(2) | 4.2 BSD |
| Reliable signals | sigvector(2)<br>other calls | 4.2 BSD |
| Shared memory for high-bandwidth communication | shmget(2)<br>shmctl(2)<br>shmop(2) | System V |
| Other interprocess communication and synchronization facilities | pipe(2),msgop(2)<br>msgget(2),msgctl(2)<br>semget(2),semctl(2)<br>semop(2) | System V |
| Asynchronous I/O for increased throughput | ioctl(2) flags,<br>select(2) | HP and 4.2 BSD |
| Synchronous I/O for increased reliability | fcntl(2) with<br>O_SYNCIO | HP |
| Preallocating disk space | prealloc(2) | HP |
| Powerfail recovery | signal(2) with<br>SIGPWR | HP and System V |

*Real-time functionality in HP-UX.*

FIGURE 3

| | Preemption Off | Preemption On | Improvement |
|---|---|---|---|
| 90% Kernel | 40 ms. | 1.4 ms. | X28 |
| 99% Kernel | 129 ms. | 3.4 ms. | X37 |
| Max. Kernel | 1127 ms. | 14.6 ms. | X77 |

*Non-preemptive kernel time (milliseconds).*

*user process can set a flag to inform the driver that it should cause the I/O request to return immediately if the request cannot be performed without blocking the user process.*

- *The SIGIO facility: Before launching an I/O request, a user process can set a flag to enable the driver to send the SIGIO signal to the process when data has arrived in the driver's input buffer.*
- *The* **select(2)** *facility: A user process can call* **select(2)** *to check whether an I/O request should be issued to one or more devices. The driver sets a bit in a user-supplied bit mask for each file descriptor that the user asked about and on which I/O can be performed.*
- *The FIONREAD facility: Before launching a* **read(2)** *request, a user process can ask the driver to tell it how many characters in the driver's input buffer are available for reading.*

These facilities can be used individually or together. For example, suppose you want to read from several terminals and you aren't sure which terminal will send you data or when to expect this data, if any. You don't want to launch a series of **read(2)** requests to each terminal, because you might end up missing data from one or more terminals as you try to read from some terminal that will never send you data.

Instead, you could enable the SIGIO facility for each terminal so that each can inform you when data has arrived in its input buffer. When SIGIO is sent, you could call **select(2)** to find out which terminal(s) are ready for reading.

A REAL-TIME APPLICATION sometimes prefers to do synchronous I/O operations to make sure that its I/O request actually completed. In synchronous I/O, a process initiates an I/O request and then suspends until it completes.

As mentioned above, the file system normally does asynchronous writes, which means that a **write(2)** returns when the data has been written only to the buffer cache, not to the disk. The data is written from the buffer cache to disk later, while the process continues to execute.

Although this asynchronous disk write increases your process's throughput, the disadvantage is that you can't be sure that your data actually has been written to disk. Therefore, HP-UX provides a flag called O_SYNCIO that lets you perform a synchronous disk write. This ensures that your data actually was written to disk.

RECOVERY FROM A POWER failure is important to real-time applications that can't afford to lose current data I/O or miss transactions. When power fails, HP-UX saves the CPU state and flushes the data cache to battery-backed-up memory. When power is restored, all I/O devices are reset, the CPU state is restored, the cache is reinitialized, I/O transactions in progress at the time of the power failure are restarted if possible, and a signal (SIGPWR) is sent to each process informing it of the power failure. Each process then can take appropriate recovery actions.

*Figure 2* Summarizes the functionality that was added to HP-UX. It presents the system call associated with the particular feature and the origin of the system call (either System V, 4.2 BSD or HP).

The performance-tuning that HP has implemented in HP-UX on the Model 840 is as important as the added real-time functionality. The following features, kernel preemption, fast file system I/O and miscellaneous performance improvements comprise the main part of HP's performance-tuning efforts.

## Kernel Preemption For Faster Response Time

AN IMPORTANT REQUIREMENT for a real-time system is quick and deterministic response time. One of the main concerns about the real-time capability of UNIX systems is that a process can execute in kernel mode for long periods of time (more than one second) without allowing a higher priority process to preempt it.

Instead, the process keeps executing in kernel mode until it blocks or finishes, while the high-priority process must wait. (A process executing in user mode gets preempted much more quickly.)

HP-UX on the Model 840 solves this problem by implementing a preemptable kernel. At certain safe places in the kernel called "preemption points" or "preemption regions," HP-UX keeps kernel data structures at a consistent state, so a higher priority real-time process can get control of the CPU at that point.

Kernel preemption is "on" at all times and is invisible to user processes, but it only affects other processes when real-time processes are executing.

The goal of implementing kernel preemption was to decrease significantly the amount of time the kernel executes before it gives up the processor to a waiting higher priority real-time process.

Known as process dispatch latency, this time was measured with a set of software tools that captured stack traces at every preemptable point, along with the time since the last preemptable point. The data was used to add preemption points and regions until the typical measured time was less than a millisecond and the maximum measured time was a few tens of milliseconds (see *Figure 3*). In other words, the improvement in process dispatch latency was 20 to 100 times better than without kernel preemption. (These measures depend on workload.)

## Fast File System I/O

FAST FILE SYSTEM performance is important to real-time applications that log data to disk files or read the data logged by other processes. For many of these applications that need quick file access, the traditional file system

of UNIX is not acceptable because:

- *Data blocks often are scattered randomly throughout the disk, resulting in large disk seek times for sequential reads.*
- *The data block size is 512 or 1024, which can be inefficient for large read and write requests.*
- *There is only one superblock, which, if damaged, can make recovery of the file system very difficult or impossible.*

The HP-UX file system has adopted its solution from the McKusick or 4.2 BSD file system. Two important features in the HP-UX file system are the implementation of cylinder groups, which reduce file seek time and add reliability, and the addition of *two* block sizes, which allow increased speed without wasting space on small files.

The HP-UX cylinder group organization reduces seek time, because many or all of the data blocks of a given file are on the same cylinder. The file system is composed of one or more cylinder groups. Each is similar to a self-contained file system, as each contains a superblock, a contiguous area of inodes and a contiguous area of data blocks.

The data block allocation policies attempt to allocate space from a given file on the same cylinder group, while placing unrelated files in different cylinder groups. Thus, many or all of the data blocks of the file are on the same cylinder, which reduces disk seek time.

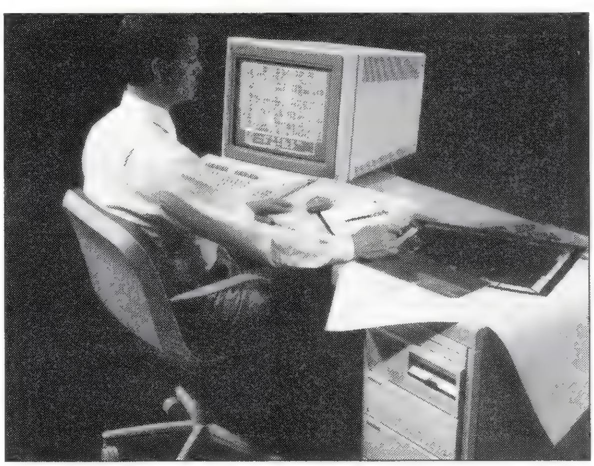
A second advantage of using cylinder groups is that having a superblock in each group means redundant copies of the superblock are maintained in case a disk read crash occurs. Also, each superblock on a particular cylinder group is allocated in such a way that destruction of all the copies of the superblocks will not occur if a single disk platter or cylinder is damaged.

The HP-UX file system uses a hybrid block size to deal with the time and space tradeoff of big versus small blocks. There is a block size that is 4 KB or 8 KB, and a fragment size that is 1/8, 1/4 ,1/2 or the same size as this block size.

Large file I/O requests are allocated and accessed a block at a time, smaller requests a fragment at a time. The block and fragment size, along with other file system parameters, can be set by the superuser at file system creation time.

The file system of the UNIX operating system normally allocates file space only as data is written. However, HP-UX allows users to preallocate file space before ever writing, either programmatically with HP's **prealloc(2)** system call or interactively with the **prealloc(1)** command.

Although this preallocated space is not necessarily contiguous, it is allocated in the best possible manner for sequential reading (without moving other data blocks). Also, the time required to do the initial write operations will be reduced, because the space already has been allocated.



*HP 9000 Series 800 Model 840 running HP-UX.*

HP-UX ON THE MODEL 840 is tuned for both real-time response and system throughput. Hewlett-Packard increased system throughput in two significant ways.

Benchmarks such as the Aim II Benchmark (Aim Technology) were run, along with HP-produced benchmarks, to track and improve the performance of specific paths in the operating system. Work loads were developed by HP to simulate real-time and other environments. GPROF dynamic call graph measurements were made to justify actions or increase overall system throughput.

Other measures, such as time from interrupt to driver, were measured with a logic analyzer interface to the hardware.

This systematic approach to performance-tuning led to very significant results, with many performance measures improving by a factor of two or more during product development. Also, this approach led to justifiable returns, including support for two-hand clock replacement algorithm and conversion of various kernel data structures from linear lists to hashed lists.

Along the way, HP also found a performance bug or two in the ported code, including a bug in the read-ahead mechanism of the buffer cache, which when fixed, increased maximum file system throughput by approximately 20 percent.

THE FUNCTIONALITY ADDITIONS and performance improvements described here form the foundation by which HP is enabling its version of UNIX to enter the real-time marketplace successfully. The features described are rather simple to implement, and in fact, most of them already are in System V or 4.2 BSD.

HP is working with the IEEE P1003 committee and the real-time subcommittee to help form a common standard by which any vendor can gain the needed functionality.

*—Suzanne Doughty is sales development engineer, Data Systems Division; Steven Kusmer is project manager, Information Technology Group (ITG); Sol Kavy and Douglas Larson are development engineers, ITG. This article is based on a presentation given at Uniforum, Washington, D.C.*

# There is a better way to find the rabbit with the blue hat.


## OMNIDEX.

In many data base inquiries, your system is forced to examine every record in the data base one at a time. Even when you're interested in only a small subset of data, the inquiry can take hours.

There is a better way, There is OMNIDEX.

OMNIDEX is an enhancement to IMAGE that gives you instant access to information. It provides:

- record selection across multiple fields without serial reads
- generic (partial key) retrieval and sorted sequential access
- multiple keys in masters
- keyword retrieval on textual data

Companies worldwide are now using OMNIDEX for literally thousands of applications. In a recent evaluation, several report writers were benchmarked with and without OMNIDEX. On a multi-field, multi-set retrieval, the original report took 63 minutes. When front-ended by OMNIDEX, record selection was completed in 7 seconds. The total report time, including sorting and formatting, required only 51 seconds... a 74-fold improvement in performance.

OMNIDEX by Dynamic Information Systems Corporation. The better way.

**Yes!** I'd like more information on OMNIDEX. Please send me a copy of your benchmarks and product literature.

Name________________ Title________________

Company________________ Phone________________

Address________________________________

HP 787

DISC 910 15th St., Suite 640 Denver, CO 80202 (303) 893-0335

ENTER 176 ON READER CARD

HP'S DESIGNCENTER ME 10.

**The Paul Mueller Company Designs Products Faster With The ME Series 10 CAD System.**

# FACTORY CAD/CAM SOLUTIONS

[BY LYNN PARKER]

"Designing with a computer should be easier than without one," says John Richardson, systems engineer in charge of CAD/CAM for the Paul Mueller Company (Springfield, MO). "And that's the way it is with our system."

Richardson is referring to the Mueller Co.'s new Hewlett-Packard DesignCenter Mechanical Engineering Series 10 (ME 10) running on the HP Series 9000 Model 320 workstation.

Mueller is a $60 million manufacturer of heat transfer surfaces and stainless steel tanks. Its clients include many of the more familiar names in brewing, as well as wineries, distilleries, beverage bottlers and customers in the meat and food processing, aerospace, cosmetic, pharmaceutical and chemical industries.

Both heat transfer surfaces and custom and production tanks are designed by Mueller using its five networked ME 10 CAD systems. The ME 10 is a member of a family of HP CAD products used in drafting, 2D design and solid modeling.

Mueller uses the ME 10 for both custom design and drafting of tanks, and for design updates to the standard product lines.

Since switching to the ME 10, Mueller estimates that it saves approximately 57 percent of the CAD cost each year over using its earlier CAD system. The company passes this cost savings on to its

customers, thus improving its position in a highly competitive industry.

Previously, Mueller had been using a mainframe system where each additional user on the network markedly degraded the system. This was especially noticeable when constructing CAD drawings, and resulted in slow drawing times.

As the company grew, its CAD needs grew, making more workstations and a more powerful system necessary.

The main reason the company chose HP's product, says Richardson, is that its interface was measurably easier to use. With the previous system, it took the operators months to come up to speed; designers employing the ME 10 system became proficient in only 40 hours of use.

System functions are directly accessible from the graphics tablet, and are complemented by screen menus and easy-to-follow prompts. For extra assistance in getting over the learning curve, the ME Series provides an online "help" facility.

But the factors that make up whether a system is easy to use include more than the interface — it also is related to the amount of frustration incurred by slow response times.

*Large beer storage tanks designed with the help of HP's ME Series 10 DesignCenter.*

The ME 10 has a predictable and almost instantaneous response time, even when used on a network, according to Richardson. The Series 10 macro language, used to automate much of the design process, also makes the ME 10 the system of choice.

LOW RELIABILITY WAS ANOTHER reason Mueller replaced its old system. The former system hardware was very sensitive to power surges. Drawings in progress, or changes to drawings that had not been saved, often would be lost when the power fluctuated.

The HP ME 10, on the other hand, is much more stable. There have been instances when the entire factory has gone dark for a few seconds and the HP system has retained its drawings.

A large part of the improvement in design process costs is a result of the exceptional reliability of the HP DesignCenter hardware. According to Richardson, "The system is pretty bulletproof." And when there are problems, HP's support staff response time is rapid — often within 15 minutes and never longer than one day.

One reason Mueller decided to purchase Hewlett-Packard

## [ ME Series 10 DesignCenter ]

The ME Series 10 offers many features to support **mechanical engineering** tasks:

■ Comprehensive construction and annotation capability.
■ Powerful command set to perform fast design modifications.
■ Sophisticated macro language for parametric design and design evaluations.
■ Customizability for specific applications.
■ Interfaces to other CAD systems (IGES), NC Programming & Finite Element Analysis.
■ A users interface designed to emulate traditional mechanical design techniques.

**Design features:**

■ Adaptive and variational design capabilities.
■ Powerful macro language for tailoring the system to company-specific requirements.
■ Area properties and model interrogation.
■ Hierarchical parts structure for assemblies.
■ Nongraphical associated text capabilities.
■ Data exchange and interfacing tools.
■ Local Area Network (LAN) or Shared Resource Manager (SRM) support.

**Drafting and documentation features:**

■ Support of drafting standards (ANSI, ISO, DIN).
■ Drafting units, linetypes and colors.
■ Ruler and grids to emulate a T-square and drafting board.
■ Construction geometry to assist in the design process.
■ Dimensioning which is associated to the geometry.
■ Cross-hatching to detail parts.
■ Various text fonts and geometric tolerancing symbols.
■ Layers for separation of parts of a drawing.
■ Isometric capabilities.

[ Plotting, a time-intensive and costly process, is one area where a powerful networked solution brings productivity savings. ]

software rather than another manufacturer's is HP's reputation for quality customer support. Among the advantages offered by HP is an 800 number staffed by trained software support customer representatives.

BECAUSE MUELLER HAS FIVE workstations networked together, the department can take advantage of data transfer, peripheral sharing and other network benefits. Mueller's ME 10 system is implemented on UNIX, a multiuser, multitasking operating system.

Plotting, a time-intensive and costly process, is one area where a powerful networked solution brings productivity savings. HP's system allows distributed processing so that a draftsman's workstation time is not tied up while drawings are plotted.

Designers can continue working while plotting their drawings on a server workstation on the network that is dedicated to plotting. This increases total system throughput considerably.

Lower amounts of mass storage are needed on the network because the ME 10 system requires half the storage of the previous system.

Currently configured as a distributed system, the network allows any workstation to access and submit commands to any other workstation. The various workstations routinely share model-oriented data as well as address the full range of Mueller's mechanical engineering CAD requirements.

All in all, according to Richardson, the ME 10 has proven itself as an excellent CAD solution for the Paul Mueller Company. Enough so that the company is considering adding other ME Series products in the not-so-distant future. —*Lynn Parker is a senior writer at Floathe & Associates, Inc., a high technology advertising and public relations firm based in Bellevue, WA.*

Would you like to continue to see articles on this topic?
Enter On Reader Card
yes 816 no 817

# The Brains Behind The Industrial Brawn

Small Computers And Process Automation

Computer process control and computer-directed testing are hot topics with engineers worldwide. It should come as no surprise that Hewlett-Packard's small technical computers are used widely as the brains behind some surprising industrial brawn. The Series 200/98xx family and the Series 80 are proven favorites for reconfigurable automated testing procedures. But by and large, the more rigorous and less predictable field of process control has been dominated by pricey minicomputers and higher attendant software costs that go hand in hand with "mini behemoths."

The venerable 80 models are useful in a surprising variety of applications in spite of the famous rallying cry, "too slow." Too slow, it seems, more often is an admission of a lack of programming expertise rather than an indictment of this particular model. I know of a number of instances in which Series 80 computers are able to monitor and run pilot and operational chemical processes with all the elan of equipment that hits your budget 50 times harder.

The secret, just as it always has been, is in understanding the difference between a computer and the software that runs on it. For its entire life history, Series 80 has been confused and misunderstood for this very reason.

If there's a single major obstacle to overcome in the use of HP Series 80 computers for process control, it's the lack of structured error management and GOSUB nesting control. Ask anyone who writes process control software what the blackest part of the mystic art really is and more often than not the answer is, "error handling and controlling limits."

[By Don Person]

Here is some code that can change all that and perhaps make an under-used Series 80 computer a bit more attractive.

## Structured Error Trapping For Series 80

A FAILING OF ALL HP BASICs plainly is embodied in 80 Series language. The unenhanced operating system has but a single memory location for remembering your BASIC ON ERROR GOTO / GOSUB declaration.

In runtime, a 3-byte address (known as ERGOTO) holds the Program Counter Pointer (PCP) of the line that declares a branch in the event of an exception. If an error is detected, the GOTO or GOSUB is taken. This makes even limited recursion very difficult since there's no way to remember or restore error branching outside a given subroutine.

Even the mighty subprogram must save and restore ERGOTO before and after each CALL to allow the SUB to use this severely constrained feature.

There's no branch reporting available to the programmer to allow branching to be sus-

pended and then reinstated without explicit knowledge of the branch in effect. You can see where this catch-22 leads: Straight to the chicken or egg circular reasoning society. If you have to know what branch is in effect to allow a temporary or local error trap, then your subroutines become increasingly less global in their potential application.

It would be much better if there were a way for a given unit of code to remember the error declaration in effect on entry, set a local error trap and then reset the original one, without having to know about it. This would allow very specific localized trapping without disrupting the overall schema of program flow.

If a local recovery trap couldn't handle the error, it could renew the error's active status, or possibly map it to a new value, allowing the caller to hook out the problem. Or it could elect to pass the whole situation back. The process I'm describing can go a long way towards making the code more modular. This is the type of error handler found in structured languages.

Another common process control problem area is runtime tracing or program backtracking. Suppose now that we have a family of subroutines that are free to call one another, but have an exit determined by some common case, like a real-time interrupt or an out-of-resources situation. Series 80 (and, in fact, all HP BASICs) provides no way to tell what our subroutine stack looks like at any given time. The information is there all right, but intentionally hidden by the system's designers.

In the 80s you have a RETURN stack with up to 255 levels of nesting, but no way to trace back to the sequence of returns pending. If a recovery routine could look back to see the sequence line number from which it was called without the necessity of causing an error, we could know the context from which a routine actually was invoked. (As HP BASIC is constituted, only functions like ERRL can do this, and the limitation is one layer deep.) This would let the object of a GOSUB statement know the calling line of the caller of the caller of the caller . . . . Ideally, we should be able to specify how deeply into the RETURN stack to go and examine the origin of the GOSUB as well.

For example, I've found it rather handy to be able to save and restore error branches. I wrote a function that opens serial files and locates the last used record using the binary search algorithm. I needed to be able to trap various file errors to test pointers, but since the function needs to be able to be referenced anywhere in the program that it's needed, there was no easy way to know what error trap to reselect on exit.

> The secret, just as it always has been, is in understanding the difference between a computer and the software that runs on it.

This particular bit of code bypasses the dangers of continually writing file pointers in the same physical location on a disk. With the binary search, you're assured of filling random access data files sequentially without the danger of "burning" the area of a disk where control files would have been.

Finding the next free record in a 16,000-unit file requires less than 14 seeks and has the advantage of never writing a record more than once. It highlights the utility of error branch abstraction, and is a case ready-made for structured error management. (See *Program 1.*)

Here's another example: A major process subroutine calls many different subroutines in turn, any of which may be exited prematurely due to end-of-line interrupts for I/O service or softkey branches. In I/O-intensive duty, it's very easy to get tangled up in your GOSUBs such that recovery from a major error is not possible. To avoid this, most people rely on linear coding with individual error traps.

Suppose instead you had a means to know the number of RETURN instructions pending and could re-establish that value on demand? You then would be free to execute GOSUBs and ON GOSUBS with complete immunity from this kind of problem. An otherwise fatal error exit could guarantee return to the exact place in the main program that a major event loop was called from. The code to do this is simple with the right assembly language. *Program 2* shows what we do on entry.

You're assured of "coming back alive" with the program intact and no pending RETURNS no matter how you tangled up the stack (short of overflowing it, that is).

Finally, remember that we're communicating with a temperamental I/O device. Now and then the unit times out while waiting for a handshake. An "on-error" branch could be set for each reference, but this is clumsy if there are many different statements involved. Ideally, we decide on timeout that we'll increment a retry counter variable and GOTO the line in question again.

Rather than having a different ON ERROR GOTO for each of the references to the troublesome contraption, consider this idea: A general purpose error trap is set up by an ON GOSUB. This single routine first decides whether the error is number 131 (timeout). If not, action is taken that is appropriate to the error number flagged. If it's our nefarious timeout, we look back to see what line number the error was called from. (We might have several different devices, right?) Then the error trap is saved and re-enabled when other branching control is needed.

## Program 1.

```
600 !   file$ is name / MSUS to open == maxrec is number of records
605 DEF FNBIN_SRCH(file$,nx,maxrec) !      buffer # nx open on exit
615     SAVE ERROR !              SAVE ERROR DECL of whoever called us
620     lo=1 !                        init hi & low record count limits
622     ON ERROR GOTO SICK  !           no disk- no file name found etc
624     ASSIGN# nx TO  file$ !                     try 2    open the file
625     GOTO NEW_ERR !                                          OPENED OK
631 SICK: RESTORE ERROR !                  force exit & rtn error number
632       CAUSE ERROR , ERRN !                  pass error back to caller
634 NEW_ERR: ON ERROR GOTO CANT_READ !  set error for E.O.File trap
638          hi,hmax=maxrec !               total records in this file
645 CALC_SEEK: IF hi>lo THEN nrec=(hi-lo)\2+lo ELSE CHECK_IT
650            READ# nx,nrec ; DUMI$ !      must hold largest datum
653            nrec=nrec+1
654 READ_AOK: lo=nrec
655           GOTO CALC_SEEK
660 CANT_READ: IF ERRN =33 OR ERRN =69 THEN READ_AOK !  4 RND/Ser
665            IF ERRN >72 OR ERRN <71 THEN SICK ELSE hi=nrec
670            GOTO CALC_SEEK
680 CHECK_IT: IF nrec>= hmax THEN nrec=0 !                file is full
690 DO_EXIT: FNBIN_SRCH=nrec
695          RESTORE ERROR !    restore OUTSIDE CALLER'S ERROR TRAP
700 FN END
```

## Program 2.

```
2000 MY_SUB: EXIT_RTN=RTNS?  ! a variable gets stack height
2010         SAVE ERROR      ! REMEMBER callers error decl.
2020         ON ERROR GOTO MY_ERR
========\
>   your  code goes here
========/
3000 MY_ERR:                         ! lcl error trapping
             .......................
             ! if we can't handle the error we do this
3050         IF EXIT_RTN > RTNS? THEN POP EXIT_RTN - RTNS?
                                     ! restore stack height
3060         RESTORE ERROR           ! reset old error branch
3070         CAUSE ERROR ERRN        ! lets caller's trap
                                     ! handle problem
```

## Program 3.

```
1820            BYT 241
1830 CAUSERR.   BIN                     ! CAUSE ERROR errnum
1840            LDMD R10,=BINTAB
1850            JSB =ONEB               ! get error number
1860            STBD R#,X10,ECODE_      ! move inline
1870            JSB =ERROR+             ! set the error and
1880 ECODE_     BSZ 1                   ! placeholder for error
```

After comparing the line number of the caller to a list of the line numbers where the suspect call is coded, we remove the pending RETURN instruction and execute a GOTO back to the offending source line to try it again, having first incremented the retry counter. Upon entry, too many retries can stop the program, warn of an uncontrollable timeout event or call for a process shutdown. Each successful operation concludes by resetting the retry count.

So, how do you find the origin of a GOSUB? And how do you remove the requirement to execute a RETURN so you can GOTO instead? The answer, of course, is assembly language. Let me introduce you to a binary program that detours the major shortcomings of system error trapping, and provide you with some great new programming constructs.

ESTRUC is a binary program for the HP 86/87 that I wrote to bring some semblance of order to event-controlled branching on my favorite small computer. ESTRUC lets you save error declarations on a first-in last-out stack, cause any system error at will, count or clear any or all pending GOSUB RETURNs and peel back the internal RETURN stack to find the line number of each pending return in the nest.

With ESTRUC, you can create error and event handlers just as flexible as those of any of the new structured languages, and get a leg up on Rocky Mountain BASIC, too. Here is a syntax guide to the complete binary.

**SAVE ERROR** — Statement with no parameters. If there's room on a 10-high internal stack, the current ERGOTO is saved. If the stack is full, error 26-STACK OVERFLOW is called. The current value of ERGOTO is not changed, just remembered.

**RESTORE ERROR** — Statement with no parameters. If there are any ERGOTOs on the binary program internal stack, the most recent one is removed and installed in place of whatever ERGOTO held before the call. In effect, the last ERGOTO saved is now back in effect, whatever its source. This should not be done if the last SAVE ERROR was called from outside an active SUBprogram, for instance. Trying to restore when there is no error stacking also causes error 26.

**CAUSE ERROR err_num** — Statement with one required parameter. Err_num must be an integer in the range of 0-254; otherwise, it's treated as modulo 256. For example, CAUSE ERROR 56 reports exactly as if an attempt to overfill a string were made, except that none of your storage space has been tortured to signal the information. It's not wise to cause errors 71 or 72, because they can force currently open disk buffers to be shut. Most system error numbers are fair game, and there are a few undocumented ones that have old 85 messages that you can use in your Gemini program. For instance:

```
ERROR 20, OUT OF PAPER
```

ERRORS 61-65 and 73-75 are retained from the HP 85. Unused tape errors are generally safe to recycle on the 86/7.

Avoid error 19 entirely unless you've experimented with the consequences.

The compound construction:

**RESTORE ERROR @ CAUSE ERROR ERRN** — Will pass an error one level back up to the previous error handler, provided you remembered to SAVE ERROR and declare it again locally at the start of your code segment.

**RTNS?** — A numeric function with no arguments. It returns the number of RETURNS on the BASIC RETURN stack. The result is an integer in the range of 0-255. Zero result indicates that no returns are waiting.

**RTN LINE?** (argument) — Numeric function with one numeric argument in the range of 1 to 256. This passes back the line number from which the corresponding GOSUB was taken. An argument of one specifies the most recent, two the next most recent, and so on. A zero result indicates you're asking for a return level deeper than the stack contains.

**POP** [level] — Statement with one optional parameter. "Level" may be a constant, variable or expression. POP without a parameter will remove the most recent RETURN obligation from the stack such that the next return instruction branches to the return below that, if any, and is equivalent to POP 1. If more than one return is to be popped, then POP with the number of returns to throw away is in order.

If the argument is greater than the stack height, then all returns present will be removed. A single POP with no pending returns will cause error 51, RETURN WITHOUT GOSUB. This works well in conjunction with the RTNS? function already mentioned.

## How ESTRUC Works

WITHOUT GETTING BOGGED down in the more esoteric details, here is an overview of the runtime code for each routine.

**SAVERR.** — We use the binary program (BP) base address as an index since the program is a relative one and can load at any address. We jump to a common subroutine that converts our relative stack pointers to absolute addresses and set the DRP to point to the stack index register. If the stack is full (i.e., there are 10D entries of 3 bytes each), then a stack overflow error is called and no new ERGOTO can be stored. Otherwise, we capture ERGOTO and push it on. We finish by incrementing the pointers and putting them away in a safe storage location within the program.

**RESTERR.** — This is the complement to SAVERR, except that it calls a stack error if there are no ERGOTOs to restore. Otherwise, it removes and restores the most recently added declarative branch.

**CAUSERR.** — This removes a signed integer from the stack, in this case, the error number to cause, and passes it over to the system via the SYSCALL ERROR+, which even handles the RTN for us. The actual code is shown in *Program 3.*

**POP.** — POP's parse code is unusual in that it tries to find any one of:

- *a numeric constant*
- *a numeric expression*
- *a numeric variable by reference*

If found, the appropriate runtime tokens are left in line. If not, we assume that there is no number on the parameter stack at runtime. Comparing R12's height against the system location Top Of the Stack (TOS) tells whether a number was found and passes it as a real or tagged integer. If present, we

### Program 4.

```
1970 ! POP [ rtns_2_remove ]     1 is the default value
1980           BYT 241            ! statement run & calc mode
1990 POP.      BIN              ! for any math
2000           LDB R20,=1       ! if no params arg is optional
2010           CMMD R12,=TOS ! a stack param passed ??
2020           JZR POP+         ! if pop with no params then cont
2030           JSB =ONEB        ! else get argument from stack
2040           STB R#,R20       ! copy to 20
2050 POP+      JSB =GSBCNT      ! how many gosubs in the nest
2060 ! gsbcnt sets ptr 2 to aim at proper byte of program
2070           JZR ER51D        ! cant pop  if no returns pending
2080           CMB R#,R20       ! as many or fewer rtns than
2090           JCY SUB22        ! same number or more rtns than
2100           STB R#,R#        ! make it pop all returns
2110 !  prevents statement like POP X when LESS THAN X rtns
2120 !         As this could blow away the operating system.
2130 SUB22     SBB R#,R#        !  reduce cnt
2140           LDMD R45,=NXTRTN ! address of the top of BASIC
2150 popLOP    SBM R45,=7,0,0  !  move back down 7 bytes=size
2160           DCB R20 ! each iteration of this loop mashes 1
2170           JNZ popLOP
2180 POP2      STMD R45,=NXTRTN     ! make this the next return
2190           STBI R22,=PTR2     ! count in BASIC PROG ctl
200            RTN                        !  all done
2210 ER51D JSB =ERROR+               ! report and dump stak
2220           BYT 51D          ! gosub-less return
```

### Program 5.

```
2240           BYT 0,55                    ! how many returns on
2250 RTNCNT.   BIN
2260           JSB =GSBCNT                 ! rtns with DRP 22
2270           STB R#,R36
2280           CLB R37                     ! rtn  less than 255
2290 makeRL    JSB =CONBIN                 ! system makes a real
2300           PUMD R40,+R12               ! push on the stack
2310           RTN                         ! done
```

use syscall ONEB to convert it to binary format for us. If not, we skip this step having preloaded a parameter of one.

Next, we syscall GSBCNT, which recovers the number of GOSUBs pending from the program control block and leaves EMC PTR2 aiming at the control block storage location. If the count is zero, all we can do is call error 51. Otherwise, we count down the number of returns to remove and adjust the return stack height and pointers accordingly. (See *Program 4.*)

**RTNCNT.** — This simply interrogates the PGM control block for the nesting value without changing it, converts from a one-byte binary integer to tagged BCD format for the system and pushes it on the parameter stack pointed to by R12. (See *Program 5.*)

**GOSLIN.** — This finds the line number for a return line by taking apart the correct 7-byte segment of the return stack. Each GOSUB saves the following information.

First there's CSTAT (R16), as one byte is pushed on, so we'll restore the machine state in effect on RETURN.

Next, three bytes of the absolute PCR for the GOSUB source line are pushed so we know where to resume execu-

tion. Finally, three bytes carrying the relative return address are pushed.

To find the actual line number, we need access to the fourth through sixth bytes of the sandwich at the correct level. First, we count through the stack to our desired position. Then we step over the CSTAT value and load the PCR for the calling line. The PCR points to the first item on the line, which is always the line number in BCD format. To get this, we load the PCR via the EMC, since it's a high memory address, and use this value to reload the EMC itself.

> . . . the 80s even can challenge the mighty Series 200 . . .

Next, we pull in the 3 bytes of the line number, checking to see that it is a legal tagged integer. We force tagged integer format by placing an octal byte 377 "tag" in front of the load and push it on the R-12 stack. ZZ0 is an option that is used to flag bad arguments or an argument out of range.

Just when you think you know the limits of a small computer's operating system, you'd be surprised at how much can be done to jam in some more horsepower. Use of 80 assembly language is a great case in point. In spite of the handicap of running with less than a 1-MHz clock, assembly language fitted carefully to available system routines can astound you with performance superior to that of much more expensive systems.

Yes, the 80s even can challenge the mighty Series 200 on almost any count. It's just a matter of knowing where to look. The reward for time well spent can be more cash left in your budget for all the things that better software coding can't buy.

In addition to the source code fragments presented here, I have the example binary search BASIC function, the assembler source code and the object binary. For details, write to: Don Person c/o AGW, Box 3103, Albany, NY 12203. *—Don Person is an independent consultant based in Albany, NY.*

Would you like to continue to see articles on this topic?
Enter On Reader Card
yes 804 no 805

# THE ANATOMY OF *Maestro*

**Developing A True Distributed Processing Application.**

**DISTRIBUTED PROCESSING**

**Michael A. Casteel**

In early 1984, we at Unison Software began developing *Maestro*, a batch job controller intended to support a network of HP 3000 systems. *Maestro* wasn't just to run independently on each of those systems but needed to continuously monitor and coordinate processing on all the different computers. Logically, it would be considered a single, distributed application.

Hewlett-Packard's DSN/DS facility is the obvious basis for such an application and we set out from the start to design this system using DS. The first thing we discovered was that the DS manual contained little or no information that would help us design a real distributed application. Its main focus was how to access remote computers while sitting at a terminal. For the sake of other users who make the same discovery, this article summarizes some of our recent experiences in distributed application design.

The focus of this article is the architecture that we developed for our application, and some of the reasoning behind it. The controlling factors are specific application parameters such as database inquiry and update profiles, transaction volume, and response time requirements. The parameters for *Maestro* are presented in *Figure 1*. It's clear that different applications would require changes to some aspects of the design.

Sadly, the present DS manuals leave the application systems programmer as much in the dark as the designer when it comes to actually *using* DS.

THE APPLICATION described here is *Maestro*, a multiple-CPU batch job scheduler and controller developed by Unison. The function of this application is to determine which batch jobs are to be run each day on each CPU in the network, then initiate and manage the processing of those jobs. Because the initiation of one job may depend on the successful completion of another job that runs on a different CPU, it's necessary for *Maestro* to monitor processing on all CPUs in the network together.

Conceptually at least, *Maestro*'s database is quite simple: Identify which jobs are running, have been run or need to run. The obvious transactions against this database, such as **Job Initiated** or **Job Succeeded**, are not likely to occur in very high volume, probably no more than a few hundred per hour, even over several CPUs.

Because it isn't an online application, it doesn't matter very much if the response time to these transactions is several seconds, which is a good thing, since network performance is hard to guarantee. Some of the principal design considerations with respect to distributed processing were:

- *Where does the application program run?*
- *Where does the database reside?*
- *How do the transactions make their way through the network?*
- *How does the application view the network environment?*
- *What happens when something breaks?*

WE QUICKLY DETERMINED that the *Maestro* control program should run on every CPU in the network. If the control program were to run only on a central computer, relying on remote access to other CPUs, a shut DS line would stop processing on the remote CPU.

# Announcing the end of an error.


*NSD announces no more hidden production errors. No more wasted effort reviewing every $STDLIST at the end of a run. And no more job listings buried in printouts. Now there's a simple way to save your time and your money.*

Our new improved version of the SMP program is here. It's called JobRescue.™ As soon as you try it, you'll understand why. JobRescue runs in the background and continuously examines $STDLISTs. If you choose, it notifies you of errors by displaying a warning on the console. You can then send this warning to other users.

With JobRescue, there is a lot more at your command. For example, now you can keep your $STDLISTs on line for more than 24 hours. Let's say you come in on Tuesday. You can review Monday's reports or hold them for another day. JobRescue lets you hold your $STDLISTS with their flagged errors for up to 127 days!

What's more, you can now specify which $STDLISTs and which errors on those $STDLISTs you want to examine. You can store $STDLISTs without errors in a file or purge them completely. Users can specify that certain $STDLISTs should be printed even if they have no errors.

With all that JobRescue does, it might surprise you how easy it is to use. You simply STREAM a job. In many cases, online commands are only one word long. In fact, because of your familiarity with HP programs, you already know most of the commands. And we guarantee JobRescue for one full year.

To show you just how well JobRescue performs, we're offering a free trial run. The end of an error means the beginning of smooth operations for you.

NSD

Looking for Trouble
HP 3000 Software

JobRescue is a trademark of NSD, Inc.

ENTER 103 ON READER CARD

Free Trial Run!
800.538.3818, (In CA) 800.538.9058
Canada 800.445.3818, Local 415.573.5923
NSD, Inc. 1670 South Amphlett Blvd.
Suite 103, San Mateo, California 94402

This would be especially onerous in a switched network, where several remote CPUs might be connected by dial-up, but cannot all be connected at once. In addition, failure of the central CPU would stop processing on every CPU in the network.

By running a copy of the application on each CPU, the programs can continue to function even if communications with the rest of the network break down. This also gives *Maestro* local access to MPE on every CPU, which helps it do its job, that is, to track all the batch jobs in execution on that CPU and to initiate (STREAM) new jobs.

BY SIMILAR REASONING, we chose to maintain a local database on each CPU in the network, as shown in *Figure 2*.

All the jobs to run on a CPU are contained in the database on that CPU, and can be initiated and tracked by the *Maestro* program on that same CPU. Again, this means that a CPU that is isolated by some fault in the network can continue to function. But what happens when jobs on this CPU are dependent on jobs run on another CPU, or vice versa?

Here is the main "Distributed Processing" aspect of this application: interdependencies between jobs that run on different CPUs. A job that runs on CPU A, for example, may need to wait until a particular job has completed on CPU B. This means that the *Maestro* program on CPU A needs to know the fate of a job that ran on CPU B, in order to decide whether to run one of its own jobs.

According to *Figure 2*, the program on CPU A would have to read the database on CPU B and check the status of the prerequisite job. If it has completed OK, the job can be initiated on CPU A. If not, the program will have to check again later, perhaps repeatedly.

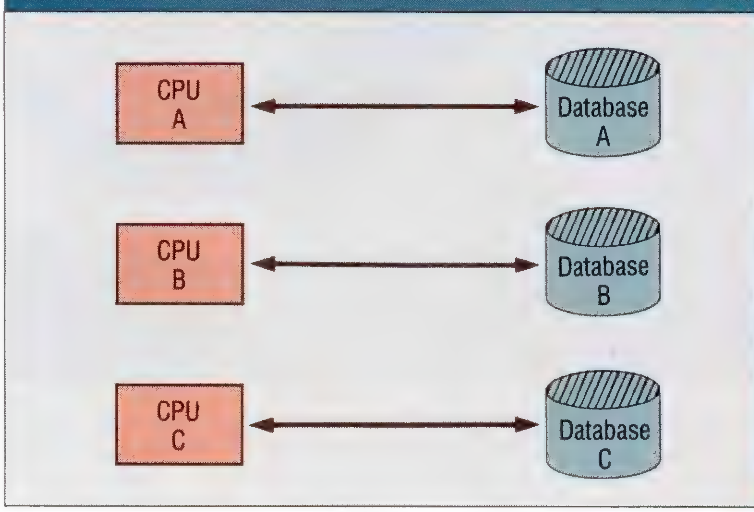
All in all, the program on CPU A may have to look at the databases on each of the other CPUs again and again, often enough to avoid undue delays in initiating dependent jobs.

FIGURE 1

| | |
|---|---|
| **Database:** | Jobs running<br>Jobs to be run |
| **Transactions:** | Job initiated<br>New job<br>Job complete<br>State change (INTRO, WAIT, EXEC) |
| **Volume:** | Moderate (100/hour?) |
| **Response:** | Flexible (10 sec?) |

***Maestro Profile.***

FIGURE 2




***Distributed Database.***

Instead, we decided to copy the entire database onto each CPU. For example, the database on CPU A will contain a record for each job on *every* CPU. When *Maestro* needs to check the status of a particular job, such as the one on CPU B, it simply looks up that job in its local database. In this way, any number of inter-CPU dependencies can be resolved quickly.

A side benefit of this approach is that the operator can use a terminal on *any* CPU to display the status of the jobs on every CPU, simply by looking in the local database.

This scheme implies that updates to the database must be posted to every CPU. When a transaction such as **Job Complete** occurs on a CPU, that transaction must be sent to every CPU, updating every database. *Figure 3* illustrates this situation for a transaction occurring

on CPU B. While each CPU reads only its local database to control processing, CPU B is shown posting its update transactions to every CPU in the network.

IN ORDER TO COMPLETE *Figure 3*, we really need to add six more arrows: CPUs A and C also will be producing transactions, and they, too, must update all three databases. The complexity of

> *. . . only MAILMAN is knowledgeable about the network . . .*

the resulting picture makes it clear that, to avoid hopeless confusion, networking considerations somehow must be separated from the application proper. Therefore, we designed a separate program, MAILMAN, to be responsible for all inter-CPU communication.

*Figure 4* illustrates his scheme: Only MAILMAN communicates with other CPUs and, therefore, only MAILMAN is knowledgeable about the network structure, or even the existence of other CPUs. In the spirit of modularity, only the application program (BATCHMAN) is knowledgeable about the database structure and the transactions that are processed against it. The only complication to the application program is that it must send transactions to MAILMAN for posting to other CPUs (if any), and receive transactions from MAILMAN which originate on other CPUs.

The details of this operation are shown in *Figure 5*. The five elements shown in this diagram are present on every CPU in the network:

- *The MAILMAN program.*
- *The application program, BATCHMAN.*
- *The application database.*
- *MAILBOX (an MPE Message File).*
- *INTERCOM (another Message File).*

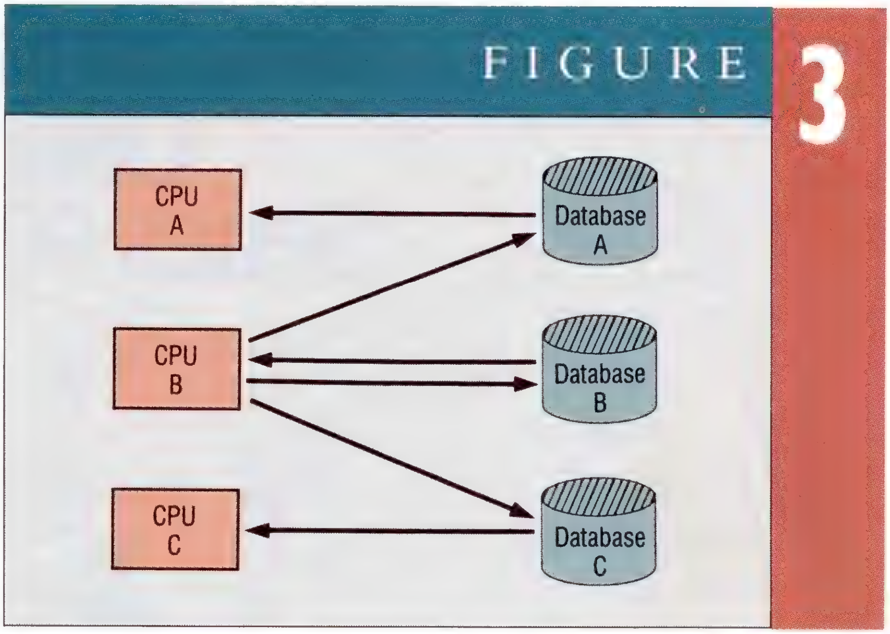
The MAILMAN is linked to other CPUs in the network using DSN/DS. In our application, there is only one copy of the application program on each CPU, so BATCHMAN executes as a son process to MAILMAN.

The application program, BATCHMAN, writes every transaction occurring on this CPU to the MAILBOX file. When the local MAILMAN reads this, it uses Remote File Access to forward them to the MAILBOX files on all remote CPUs to which it is linked, so that their databases can be updated accordingly.

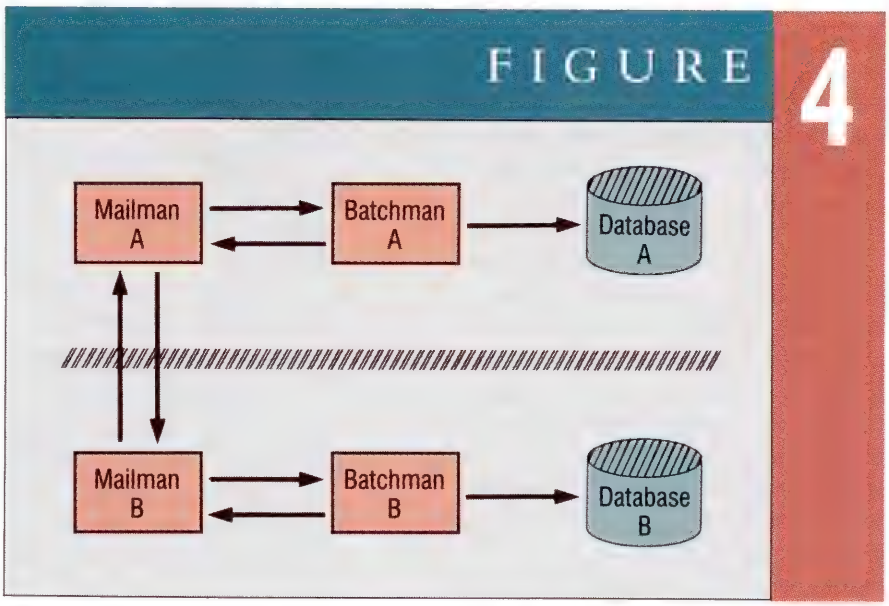
Similarly, the MAILMAN programs on other CPUs also use Remote File Access to write their transactions to the MAILBOX file on this CPU. When MAILMAN reads these, it sends them to the local application program via the INTERCOM file. This allows the local database to be updated in accordance with transactions occurring on other CPUs.

In addition, MAILMAN forwards transactions received from a remote CPU to the MAILBOX files on other remote CPUs. As will be seen below, this inter-CPU message forwarding is useful in reducing the degree of interconnection, together with its associated overhead.

Note that by having the local BATCHMAN and remote MAILMAN programs all write to the same local MAILBOX file, the MAILMAN has only one input file. If MAILMAN had to read its input from multiple input files, it would complicate the situation significantly.




***Distributed transactions.***

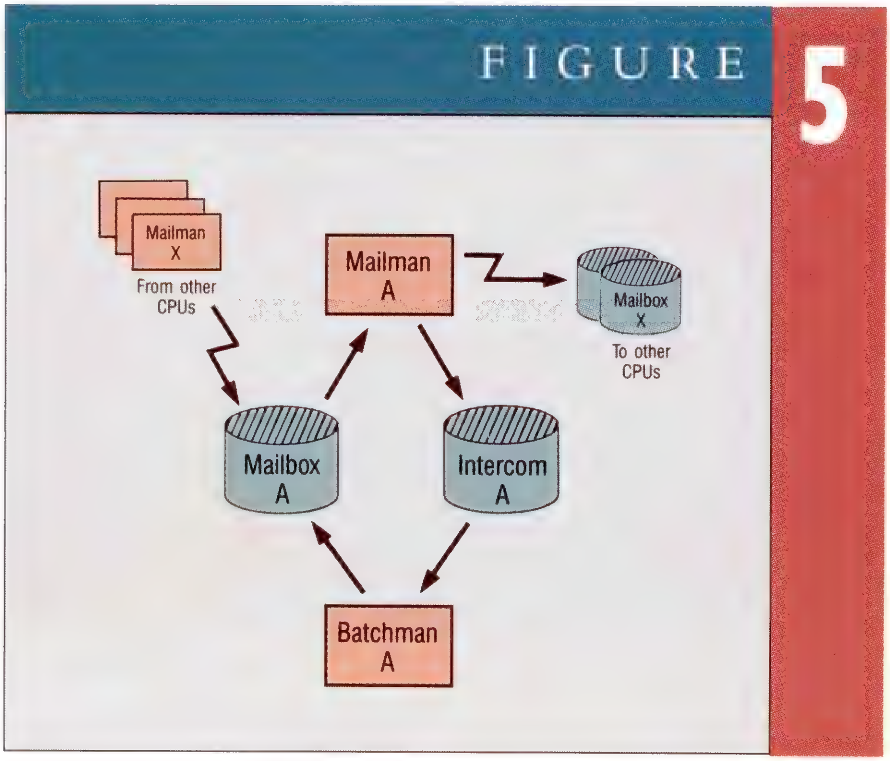



***The MAILMAN network interface.***

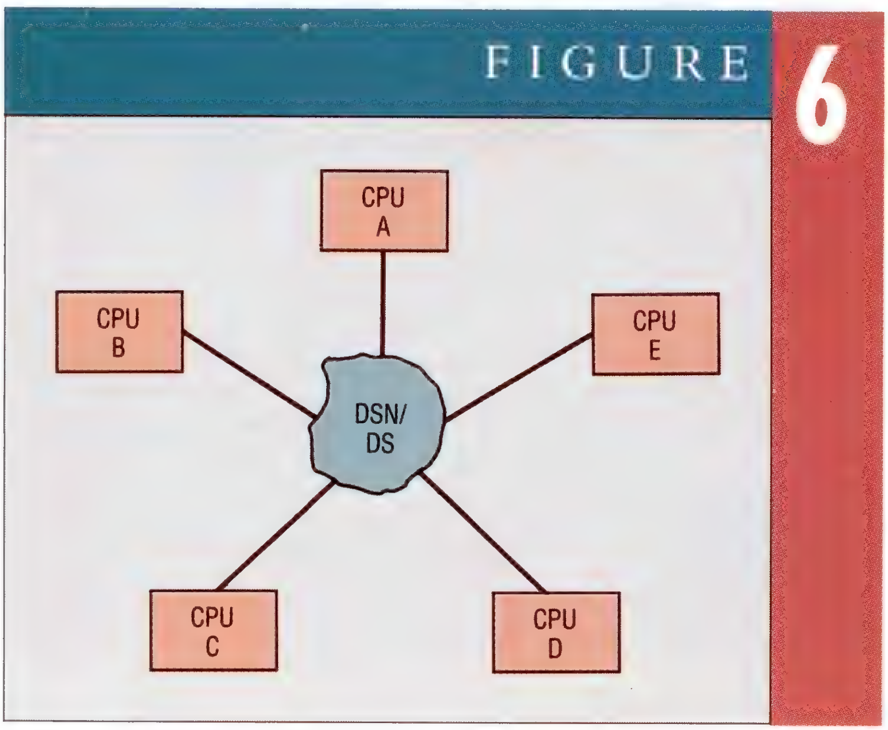
A TYPICAL REPRESENTATION of a computer system network is shown in *Figure 6*. A simplistic application of the scheme developed so far would have the MAILMAN on each CPU link to each of the other CPUs. For example, CPU A in *Figure 6* would link to CPUs B, C, D and E; CPU B would link to A, C, D and E; and so on.

Under Hewlett-Packard's DS, this means that each MAILMAN would create remote sessions on (in this example) four CPUs. Conversely, each CPU would host four remote sessions from the remote CPUs, just to support this one application. In general, if a network contained "n" CPUs, then each CPU would host "n-1" remote sessions. It seems likely that a large network of this sort would require an excessive amount of DS resources.

With this in mind, and being somewhat distrustful of this high degree of interconnection anyway, we instead have imposed the logical network structure shown in *Figure 7*. One CPU (CPU A in this figure) is designated the "primary" CPU, and the other CPUs are "sec-




***Communicating transactions to the network.***




***Conceptual network layout.***

ondary." The "cloud" network of *Figure 6* is replaced with the "star" of *Figure 7*. The primary CPU links to each of the secondary CPUs, but the secondary CPUs only link to the primary. This means that only the primary CPU still hosts "n-1" remote sessions; each of the secondary CPUs now hosts only one remote session for this application.

*Note:* This is only the LOGICAL network structure recognized by *Maestro*, not the physical. We have found that it is ordinarily possible to realize such a logical network structure under DSN/DS regardless of the physical structure, and quite simple to do so under new DS products, such as X.25 and LAN.

The MAILMAN on the primary CPU sends transactions originating on it to all the secondary CPUs directly, since it is linked to each one. The MAILMAN on a secondary CPU, say CPU E in *Figure 7*, only sends its transactions to the primary CPU. However, the primary MAILMAN forwards these transactions to each of the other CPUs, so they receive the transactions in any case, even though they are not linked directly to the originating CPU.

As mentioned earlier, the presence of the *Maestro* application and database on every CPU in the network is an advantage in the event of a component failure. Should some CPU in the network fail, the other CPUs can continue processing. Of course, if some jobs are dependent on jobs that were to run on the failed CPU, there could be a problem. In such case, our scheme of posting updates to all CPUs has the added advantage that, if the job already ran on the failed CPU, all the other CPUs already know it, so a subsequent failure of that CPU need not hold up their processing.

Our experience has shown that Murphy's Law is the rule in a distributed processing network: If anything can go wrong, it will; and, since there are so many things to go wrong (CPUs, DS links), it seems that something is always broken. Or, if we're lucky and everything is in working order, there are still occasions to shut down a CPU (to

UNISON Software
415 Clyde Avenue
Mountain View, CA 94043
(415) 968-7511
**Enter 604 on reader card.**

update MPE, for example) or DS line (perhaps to dial up a different CPU). What this means is that, when a transaction is generated (**Job Succeeded**), we can't count on being able to send this transaction to any other CPU, since some element in the path to that CPU may well be broken.

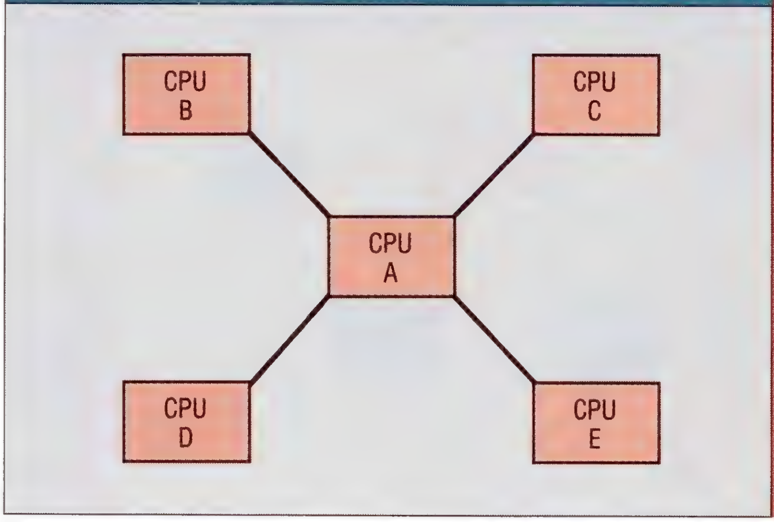
THE FAILURE OF EITHER a CPU or DS link can disrupt communications between MAILMAN and a remote CPU. We did not wish transactions to be lost due to such an interruption, which may be only temporary. In fact, we felt that it was desirable to allow communications to be stopped and restarted at any time.

We devised the following mechanism, illustrated in *Figure 8*, to preserve the integrity of the application in such events.

Referring to *Figure 8*, the MAILMAN on CPU A normally sends transactions to the MAILBOX file on CPU B via Remote File Access. If either the DS link or CPU B fails, or the link is shut intentionally, MAILMAN A will be unable to write to the MAILBOX on CPU B. It then will begin writing the transactions to a local file, which we have called a "POBOX." Every CPU contains a POBOX for each CPU to which it is linked. Secondary CPUs hold only one POBOX for the primary CPU. On the primary CPU, there is a POBOX file for each of the secondary CPUs.

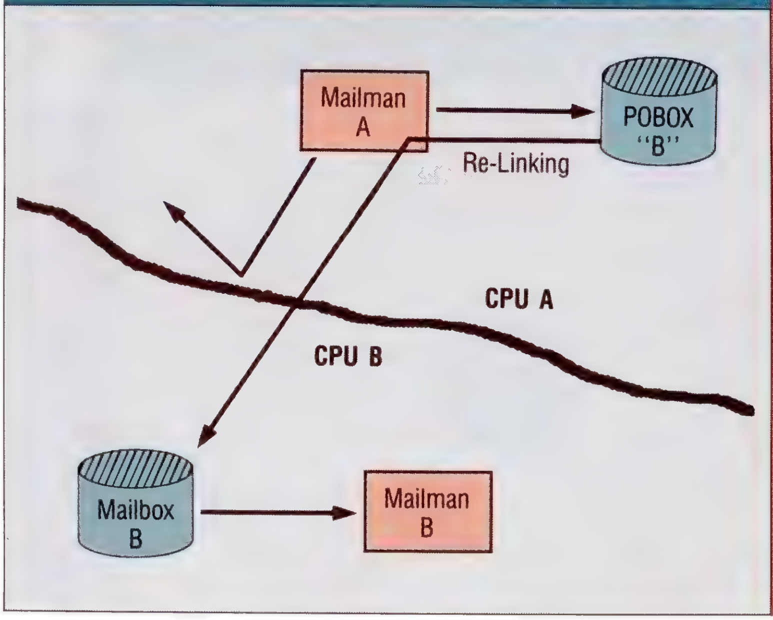
When communication is re-established with the remote CPU, MAILMAN first copies the contents of the POBOX to the remote MAILBOX, then resumes writing transactions directly to the remote MAILBOX. Note that this scheme preserves the chronology of the

FIGURE 7




**Maestro *logical network.***

FIGURE 8




***CPU A unlinked from CPU B.***

transactions as well as the transactions themselves.

The "star" network design raises a different problem if the primary CPU fails, since there is then no communication among the remaining CPUs. This problem isn't a severe one, however, since the designation of the primary CPU is essentially arbitrary. If communication links permit, it's a fairly simple matter to designate another primary CPU and restore full communications.

*Murphy's Law rules in a multiple-CPU network.*

THE DISTRIBUTED PROCESSING scheme described in this article has been in operation since December 1984. A number of implementation details have evolved since then, as we learn from experience about operating in the DS environment, but this architecture has proven to be adequate for our application.

It's clear that the possibility of component failure must be taken into account in the system design. Although the HP 3000, MPE and DS generally are reliable systems, Murphy's Law rules in a multiple-CPU network. Things are going to break, and the software had better be able to cope with it.

To date, the *Maestro* architecture has proven quite robust, even in the face of MPE system failures and DS hang-ups. Damage is ordinarily restricted to the CPU or link which failed, while other CPUs continue to process normally. Full recovery is usually accomplished simply by restarting the failed CPU or DS link.
*— Michael A. Casteel is cofounder and executive vice president of Unison Software, Mountain View, CA.*

Would you like to continue to see articles on this topic?
Enter On Reader Card
yes 810 no 811

# A BOLD APPROACH TO SPECTRUM

**_Speedware_ Users In For Exciting Times.**

**SPECTRUM**

**Karen Heater**

*Editor's Note: This is the second part in a series of articles based on third-party vendors' experiences in migrating to the HP 3000 Series 900 (Spectrum). HP's recent commitment to August shipments of the Series 930s (at a lower-than-expected price) and placement of the Series 950 in active order status (see "News & Trends," this issue), third-party vendors and customers alike gear up for an exciting second-half 1987.*

Infocentre Corporation is a leading supplier of HP-based software solutions around the world. The *Speedware* environment is the company's most well-known product. This is a complete family of fourth-generation development and run-time software, the bulk of which is written in SPL.

"SPL was chosen for performance reasons," says Ray Ouellette, product manager. "*Speedware* was designed to be a high-performer at run time and there is just no choice on the standard HP 3000s with their stack architecture — SPL is it."

"When we developed *MicroSpeedware* (our MS-DOS version of *Speedware*), SPL was not an option, so C was used. We took the approach of developing an SPL to C converter program, so when HP conceived Spectrum, we already knew which language we would be using for the future."

Infocentre was one of the first companies on HP's Fast Start Program and had early access to Spectrum. However, the company ended up taking a more radical and bold approach to the project.

Ian Farquharson, company chairman, explained: "The pressure was on from a marketing perspective. Everybody went crazy trying to rush products out the door that they could claim operated in 'native mode.' To get *Speedware* in C, and therefore native mode *fast*, would mean another SPL to C conversion, much as we did for MS-DOS. We decided against that approach."

INFOCENTRE'S EXPERIENCE in conversions had shown that, although one gets from point A to point B faster, the finished product carries with it many design features of the original, which may not suit the new environment.

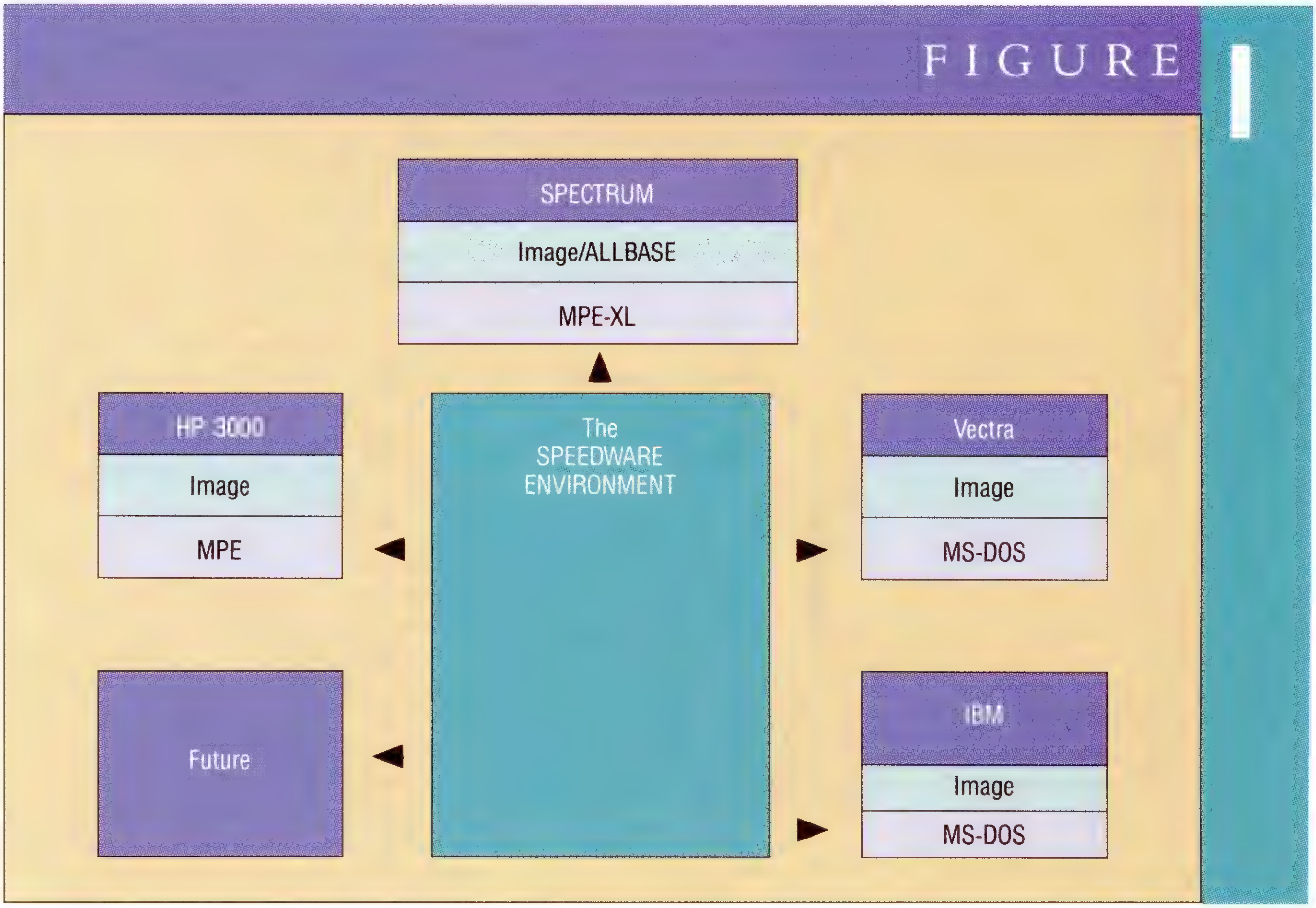
Infocentre was impressed with Spectrum's design and, being committed to performance, they wanted to take advantage of these features in *Speedware*. A conversion wouldn't do that automatically.

"We thought our customers deserved better," says Farquharson. "They expect ongoing innovation from us anyway, and there's a lot in Spectrum to take advantage of."

Infocentre embarked on a two-step approach. The first, rather short step, was to ensure clean, reliable, compatibility mode, followed by the conversion of some much-used routines to C (in native mode) to ensure the kind of performance customers would expect. The first step was to be ready for customers the day Spectrum finally was shipped.

The second step was of a much larger scale: Develop the *next generation* of *Speedware*.

Getting much information on this project is difficult due to its confidential nature, but President Jean-Pierre Theoret confided, "Although the *Speedware* environment is already quite advanced software technology,




we felt that the time was right to develop for the future.

"The industry went from 3GL to 4GL. We now are moving from 4GL to a complete environment, the 'fourth-generation environment,' consisting of functionality well beyond a mere language. Infocentre has decided to use the Spectrum opportunity to take our current *Speedware* environment to whatever that next technological step may be. A product for the next 10 years."

Citing Spectrum as an example, Theoret pointed out that systems of the future will have massive amounts of addressable main memory and almost limitless disk capacity. However, PCs and workstations will be more prevalent than ever and the need will grow to develop applications in PC/mini/mainframe networks.

Such networks will be complex, with different hardware, operating systems and communications protocol (e.g., Spectrum with HP 3000, with IBM PCs and VAXs running everything from MPE-XL to MS-DOS and UNIX — and who knows which LAN!).

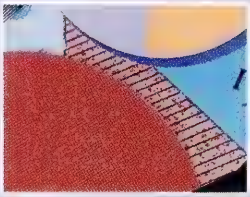
"Our objective," said Theoret, "is to better the system developers from this barrage of technology, with the *Speedware* environment. Yet at the same time, take advantage of it all."

THE INFOCENTRE PROJECT, it would seem, is very comparable to HP's decision to do Spectrum: A big project based upon new foundations for the hardware and operating systems of the future. Like HP, it is committed to compatibility and upgrade paths for its customers.

Infocentre certainly knows and understands what HP has gone through with its Spectrum project. "At the end of the day," says Farquharson, "HP will deliver a product that demonstrates to the world why its customers remain so loyal and patient. They trust HP. To take quantum leaps you have to make brave decisions." —*Karen Heater is corporate marketing manager at Infocentre Corporation, Mississauga, Ontario.*

**Infocentre Corporation**
7420 Airport Road
Suite 201
Mississauga, Ontario
Canada L4T 4E5
(416) 678-1841
**Enter 607 on reader card.**

Would you like to continue to see articles on this topic?
Enter On Reader Card
Yes 812 no 813



**DBMS**

**David Merit**

**Squelching The 'Half-Truths' About HP's Efficient DBMS.**

# IMAGE Performance Myths

"Myth" may be too strong a word to describe the commonly-believed taboos about the IMAGE and TurboIMAGE database management systems on the HP 3000. Perhaps a better term would be "half-truth."

Regardless, praises of integer-keyed master, non-prime capacities and sorted paths send users into choruses of "but we've been told they're bad." Unfortunately, for the most part, users have not been told the whole story.

There are both good and bad points about the various facilities available in IMAGE. Without examining their nature and characteristics, it's easy to arrive at the wrong conclusions about their purpose and worth. Likewise, without looking at the reasons for some of the so-called rules about IMAGE performance, it's difficult to maintain your databases at optimum performance.

### "Secondaries Degrade Performance"

Secondaries (also called synonyms) are an unavoidable by-product of the very efficient method of hashing IMAGE uses when adding entries into and retrieving them from master datasets.

This method is designed to add entries to and retrieve them from master datasets with one disk I/O. (Actually, at least two I/Os are required for DBPUTs, but one only decrements the dataset's free-space counter, which is not a function of the add mechanism per se.)

IMAGE places entries into master datasets by hashing the value of the search field of the incoming entry into a record address and then storing the entry at that location. To later access the entry, IMAGE hashes the key value you supply through the same algorithm and looks for the entry at the calculated address. This method normally provides very quick addition and retrieval of entries.

One drawback of hashing is that multiple entries can hash to the same location, since it's improbable that the hashing algorithm always will calculate unique addresses.

*There are both good and bad points about the various facilities available in IMAGE.*

When an entry hashes to a location already occupied by a primary entry (one residing at its calculated address), IMAGE places it elsewhere in the dataset and chains it to its primary. IMAGE attempts to place the secondary entry in the same block as the primary. If there is no free space in the primary's block, IMAGE places it in an adjacent block.

If the incoming entry hashes to a location occupied by a secondary, IMAGE relocates ("migrates") the secondary and places the primary at its vacated location. A condition called clustering, in which many entries hash to the same locations, can require even more I/Os to add new entries, because IMAGE may have to read many blocks before it finds a free location.

This method requires that entries be relocated only when a collision occurs: IMAGE's average unique-hit rate is about 70 percent (for a master dataset, which is about 80 percent full), so entries usually are added with two I/Os and retrieved with one. This hashing method is preferable to other access methods, such as KSAM, which always require at least two I/Os.

Even though secondaries can take more I/Os to place and access than primaries, if a secondary is located in the same block as its primary, IMAGE doesn't have to do another disk I/O to access it.

SO, SECONDARIES DON'T necessarily require additional I/Os to process. Again, it matters whether the secondary resides in the same block as its primary. There are two criteria that influence this, the dataset's blocking factor and its distribution of entries.

The blocking factor represents the number of entries in each dataset disk block, so it fixes the odds for a secondary, finding a free location in the same block as its primary. A secondary has a much better chance of being blocked in with its primary in a master dataset with a high blocking factor (say, 10) than with a low one (say, two).

Again, since all disk I/O against IMAGE datasets is done in blocks, the number of entries gotten with each I/O is equal to the blocking factor. With a blocking factor of one, secondaries always require additional I/Os. But with a double-digit blocking factor, more than one I/O seldom is needed.

The distribution of entries in the dataset — how evenly entries hash — also is important, because it determines

whether free space will be available at locations throughout the dataset. After all, even with a high blocking factor, if entries all cluster together and saturate many contiguous blocks, a potentially long serial search through blocks with no free space is required to place a secondary.

In this scenario, it can take hundreds or thousands of disk I/Os to DBPUT entries into a master dataset, and it's precisely this condition that lies at the root of the next myth.

### "Master Set Search Fields Should Not Be Integers"

As well as IMAGE users know their own names, they know never to use integer-type fields as master dataset search fields, because they can lead to performance problems. While this is true in many circumstances, if used properly, integer-keyed master datasets outperform and are more disk-efficient than those with alphanumeric search fields. The reason for caution is fundamental to the nature of the hashing.

IMAGE uses two algorithms for placing entries into master datasets: one for alphanumeric keys (data types X, U, P and Z) and one for integer keys (data types I, J, K and R). The alphanumeric algorithm calculates a location in the master dataset by first converting the search field value into bits and scrambling these bits to arrive at an integer. It then subtracts one from this integer, divides it by the capacity and adds one to the remainder.

For integer keys, since the search field value already is an integer, IMAGE performs only the latter step of the alphanumeric algorithm. This calculation results in the entry being placed at the relative record number equal to its key value. For example, a key value of "10" would be placed at relative record number 10.

If the key values are sequential numbers with no gaps, the resulting integer-keyed master dataset contains a solid block of entries with no free locations at all between then. This is perfectly efficient: Every entry is a primary, so no additional I/Os are required for processing secondaries.

Also, there can be significant disk savings, because an integer key can be represented in less disk space than an alphanumeric key. For example, an I2 takes only two words of disk, while its alphanumeric equivalent, an X12, requires six words.

But it's this solid entry distribution that causes the downfall of the dataset. The classic problem with integer-keyed masters is that, because old entries are deleted (opening up free space for new entries throughout the dataset), at some point a key value exceeds the capacity and new entries begin to hash to already-occupied locations.

Because IMAGE's method of placing secondaries is to read serially and place the incoming entry at the first free location, and since it's likely that an integer-keyed master contains large areas with nothing but solid entries, it can take many I/Os to place incoming records.

So, a good candidate for an integer-keyed master is a dataset in which key values are always less than the capacity. Examples are datasets that contain messages with related codes, accounting codes, control information and account numbers.

Master datasets that contain keys of somewhat random values should never be integer-keyed. Common examples are Social Security numbers, customer numbers and part numbers, since they either cannot be controlled, are often assigned in a non-incremental fashion or are prone to heavy additions and deletions.

### "Capacities Should Be Prime"

Because the capacity of a master dataset is an element in the placement algorithms, it follows that the capacity is significant in determining the performance of master datasets, specifically by influencing the evenness of the overall distribution of entries and thereby determining the amount of secondaries.

It's generally agreed that prime numbers make the best capacities, because they tend to break up patterns in numbers and, therefore, introduce uniqueness. Our studies at Bradmark and those of several others dispute that prime numbers make optimal capacities. In fact, we've found that a prime capacity most often results in a poor distribution, if not the worst distribution, when compared with other capacities in its vicinity.

The test results also demonstrate that the distribution of entries in a master dataset is extremely sensitive to its capacity and that capacities within even a very narrow range can result in entirely different distributions — even if the capacity is increased or decreased by one.

Since we haven't found an algorithm to determine an efficient capacity, we test various capacities using a sample of the entries in the dataset (after all, such a study would be meaningless without the other significant element in the algorithm, the key values).

Overall we've concluded that a prime number rarely will result in a poor distribution of entries. But there normally are non-prime capacities that will result in better distributions, although many non-prime capacities will result in far inferior distributions to most prime capacities. In other words, if you're plucking a capacity out of the air, a prime number is best. But if you sample various capacities, the best one probably won't be prime.

Remember, even a poor distribution of entries may be quite adequate if the blocking factor is high enough to absorb the impact of the secondaries.

### "The More Free Space, The Better"

Another significant factor in master dataset performance is the percentage of fullness in the dataset. The rationale is

that free space minimizes secondaries by helping to assure that available locations exist throughout the dataset. The general recommendation is that the capacity should not only leave room for growth, but also provide 20 to 25 percent free space.

Some go overboard and choose capacities that are double and triple the active entry count, assuming that the surplus free space will guarantee very few secondaries. But this doesn't seem to hold true. There's a threshold of free space that is beneficial: Anything beyond 25 percent free space doesn't help distribution and actually can degrade performance in other areas.

An unnecessarily large master dataset takes longer to read serially than one that has only the recommended amount of free space, because it's comprised of more blocks that take more disk I/Os to process. Additionally, it takes up more disk space and requires more time and tape to back up.

For integer-keyed masters whose key values are always incremented by one and never deleted, no free space other than for growth is needed.

### "Change Capacities At 80% Fullness"

Since free space helps to minimize secondaries, it's generally recommended to increase a dataset's capacity at 80 percent of saturation. In fact, there are a number of programs that flag datasets that meet this criteria to alert users that their capacities need to be increased.

But what if at 80 percent fullness a master dataset contains 95 percent primaries and only five percent secondaries?

The normal primary/secondary ratio in this situation is 15 percent, so it would be senseless to change its capacity and risk increasing the number of secondaries unless additional free space were required for growth.

Conversely, a capacity change at 80 percent may be too late — performance already may have started to deteriorate because of excessive collisions.

There's nothing that says a master dataset cannot have a very poor distribution at only 50 percent saturation, so a dataset may warrant a capacity change long before 80 percent. The only successful method for determining whether a capacity change is needed to improve performance is to use a diagnostic to analyze the distribution and primary-to-secondary ratio.

Remember to focus your attention on the master datasets with low blocking factors. It's in these datasets that you can ill-afford poor distribution.

### "The Sample Rules For Masters Apply To Details"

Hardly any attention has been given here to the considerations for detail dataset capacities. This is because details are very simple: Entries added to detail sets don't hash, they're appended, so you don't need to use a prime number nor leave extra free space.

The same programs that flag 80 percent full masters also flag details that are 80 percent full. It may not be a good idea, though, to change the capacities at that time, because percentages are relative and 20 percent free space may equal enough free entries to last a long time. After all, consider that an 80 percent full dataset with a capacity of 250,000 contains 50,000 free locations. Of course, an equally full dataset with a capacity of 100 contains only 20 free locations.

This indicates that you should look beyond the percentage of fullness in deciding whether a detail set requires a capacity change. Also consider the set's contents: Very dynamic information? Five years of history? A huge detail set at 99.9 percent saturation may not be the problem.

Remember that bigger detail sets require more disk space and take more tape and time to back up, serial reads take no longer regardless of the dataset

capacity since IMAGE reads only to the high-water mark (which reflects the highest active entry), and a database utility can change a detail set capacity in seconds or minutes.

### "Deleting Entries Speeds Up Access"

Many sites delete from their database inactive data (e.g., paid invoices, dead accounts) or data that has aged sufficiently. This not only frees up space in the datasets, but also can provide a noticeable difference in the speed of accessing detail entries in chained mode, since the chains are shorter.

But there's a very significant down side to deleting entries from detail datasets that counteracts the performance gains. The problem results from the way IMAGE reuses space made available by deleted entries in detail sets.

Unlike a master dataset (in which secondaries migrate to the locations vacated by their deleted primaries), deleted locations in a detail set are chained together on a singly-linked list called the delete chain and reused before newly-added entries are appended to the end of the set.

The performance problem impacts not the existing entries in the set, but the new entries that are added. Because the selected entries may have been scattered throughout the set, the new entries that take their locations also are dispersed. This degrades performance for the same reason that secondaries can degrade performance in a master: Entries that are in different blocks require more disk I/Os to access than if they were in the same block.

Normally, performance improves after such mass deletions, because the chains are shorter and processing is being done on existing entries on those chains. But as new entries are added (and scattered throughout the set) and processing shifts to those entries, additional I/Os are required and performance drops. Very often, performance becomes worse than it was before the deletions were done.

The locality of the deleted entries determines how many I/Os will be required to process the new entries that take their place. If the deleted entries are close to one another, the new entries will be, too, and may require fewer, if any, additional I/Os to access than if they were appended to the set.

*IMAGE is not a stupid database management system, and it wouldn't contain facilities that aren't useful.*

The number of deleted entries determines how long the performance problem will last. Since new entries are appended once the deleted entries are reused, the processing of these entries will be faster (although it still may take longer to access the entries that took deleted locations).

The solution to the problems caused by deleting detail entries is to circumvent the normal method of reusing their locations and force IMAGE to append entries instead. This is accomplished simply by removing the delete chain. One method for doing this is a chained DBUNLOAD/DBLOAD, which HP recommends you do on a regular basis to maintain optimum performance. There are utilities that accomplish the same task of reorganizing detail datasets faster and easier to disk instead of tape.

### "Detail Set Organizations Improve Performance"

Although reorganizing a detail dataset most often dramatically improves performance, it also can yield no performance gains and actually can make it worse.

A reorganization is accomplished via a chained unload which reads serially through the master dataset related to the detail on its primary path and unloads each chain to tape. The dataset then is erased (or purged and recreated) and then the entries are loaded back in.

One benefit is that, because only active entries are unloaded and reloaded, the delete chain is eliminated. Another is that, because all the entries are unloaded in chained order and then reloaded in the same order, the entries on the various chains are made contiguous. This is beneficial, because when the entries on a chain are contiguous, they can be accessed in less disk I/Os than if they were spread apart. So, besides eliminating the delete chain, a reorganization reorders entries so their chained access is more efficient.

So why would a reorganization not improve performance? How could it make performance worse?

Performance may not be improved by a reorganization, because the fragmentation of the chains may not be too bad and the improvement may not be noticeable; or the chains may be very short and, therefore, the entries may reside in the same blocks already. You should run a diagnostic first to determine whether a reorganization is worthwhile.

A reorganization may not improve performance and may make performance worse if you believed the next myth when building your database.

### "The Primary Path Should Be The Most Frequently Accessed"

Of the paths related to a detail dataset, one is considered to be the primary path. The assignment of the primary path is significant, because the detail entries are unloaded in the order of the chains on the primary path. Since the

entries are reloaded in that order, the primary path dictates in what sequence the entries physically will reside following the reorganization.

So, why should the most frequently accessed path not be the primary path? Actually, it should, with one reservation: It should be the path with the most *chained* access, because that's what benefits from a reorganization.

One common example of a misassigned primary path is a detail dataset whose entries have a one-to-one relationship with an associated master dataset (one detail entry per master entry), which should never have the primary path assigned to it, because that path contains no chains. This database design flaw is seen time and time again, even in third-party and HP packages.

Often, it's the common database construct in which a detail dataset contains a header record for each invoice, and the detail set is pathed to two masters. One is an automatic master that contains invoice numbers, the other is a manual master that contains customer account numbers. The primary path is assigned to the automatic master.

Again, there are no chains snaking through the detail dataset for the invoice number path, because there is never more than one header per invoice. However, each customer may have tens or hundreds of invoices, so the primary path should be assigned to the second path, even though the access along the first path may be much more frequent.

If two paths are accessed chained with about the same frequency, you should make the primary path the one with the longest average chain length. Use a diagnostic to determine the average chain lengths on each path.

Besides the possibility that primary paths were not assigned properly in the original database schema, it's also possible that a primary path will move. Beware of the technique in some database utilities of deleting a path and adding it back in to correct a broken chain by rebuilding them all. If this was the primary path, it won't be when you're done.

Probably the reason that primary paths often are misassigned is that they don't *need* to be assigned. A detail dataset definition in a schema that doesn't have the primary path specified (with a "!" next to the search item name) automatically is defaulted to the first unsorted path.

The primary path easily can be reassigned by use of a database utility or via a DBUNLOAD/DBLOAD. You can do it even if you use a third-party or HP software package. They neither know nor care which detail path is defined as the primary path.

THESE HAVE BEEN just some of many myths or half-truths about IMAGE database performance. While none is without foundation, they have been generalized to the point that IMAGE users follow them because they don't know better.

IMAGE is not a stupid database management system, and it wouldn't contain facilities that aren't useful. Some of them have gotten bad press from users who have misused them and thus authored their own horror stories. Others waste disk space and time unnecessarily.

Those who design poor database structures are at fault, not IMAGE, when DBPUTs take minutes to complete. Also at fault are those responsible for database maintenance who allow such nightmares to continue and who throw additional resources like memory and disk at the problems to try to make them go away. This works sometimes, but it's expensive at the least.

Keeping your databases at their optimum requires not only time, resources and diagnostic and maintenance tools, but also an understanding of both halves of the truths about IMAGE performance. *— David Merit is vice president, technical services, at Bradmark Computer Systems, Los Angeles, CA.*

Would you like to continue to see articles on this topic?
Enter On Reader Card
yes 800 no 801


3000



# Data Center Management

James F. Dowling

In last month's installment, we looked at the broadest aspects of a performance management program. Three system activity states (Up, Down and Available) were defined and used to present a user's view of overall system service levels.

Once we have a plan for keeping the system available, we can turn our attention to the work that's done on the machine. In my experience, much more user dissatisfaction and computer operations turmoil results when batch jobs abort than when transaction processing time increases even significantly.

Consider the following sequence of events:

1. A report writer batch job aborts.
2. Operator prints $STDLIST.
3. Responsible system analyst is notified and elects to hold the fix until morning.
4. Next morning the report recipient arrives and does not receive his/her report as expected.
5. Analyst arrives and determines that the program aborted for lack of sufficient disk space.
6. Action is taken during the day to make more disk space available.
7. The job is rerun that evening.
8. Report is delivered a day late.
9. Report recipient is not happy, but is satisfied.

This is about as clean a situation as one might expect. What if the job was an update process that interfered with the entire production schedule? The analyst would have had to go into work to fix the problem only to find out that there was nothing that could be done. The system manager would have been called in to locate some disk space.

Once the space was found, the production schedule could resume causing all of the night's work to be delivered to the users late the next morning. And worse yet, the system might have been kept from the users for several hours to finish the night's work and system backups.

Even this is a nice turn of events. It could have been and often is worse. The lack of disk space causes the print spoolers to shut down, thereby indirectly causing several other jobs to abort, making the recovery efforts much more complex, involving operations, programming and user representatives to determine the best way to recover and when to let the online systems come up.

Sometimes the recoveries aren't well thought out and problems crop up as the day and week proceed. It suffices to say that "Prevention is the best medicine," so please read on.

Lack of CPU or memory resources will make your system slow down as MPE compensates for the shortage. Lack of disk space, on the other hand, isn't handled very well at all by MPE. Print spooling stops, programs fail, system backups fail and it even may become impossible to start the machine after it is shut down or fails. Disk space is manageable and the problems resulting from lack of disk space can be avoided.

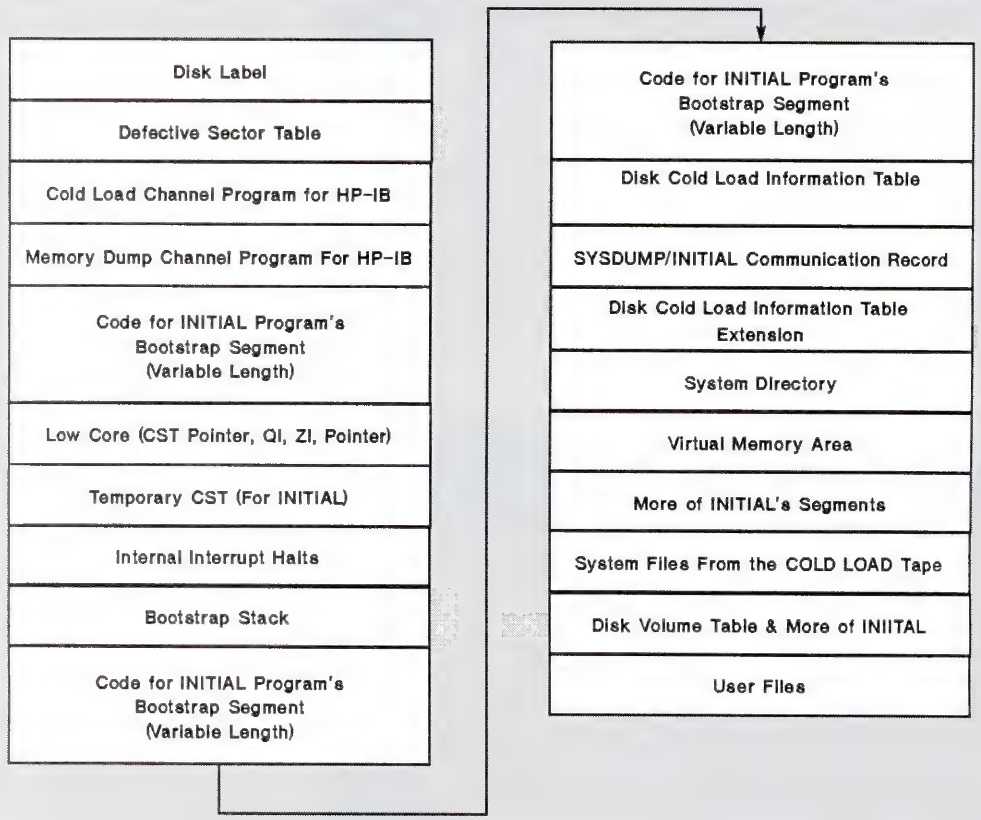
DISK SPACE IS ONE of the key resources on any computer system. Like all other critical resources, it must be measured and managed to ensure good overall system performance. To manage disk space, it first must be understood in two contexts. Disk space capacity and consumption form the measurement/management context. Disk space availability and use form the performance characteristics context.

A simple disk system consists of a disk-shaped magnetic recording surface, a read/write head, a disk rotation mechanism, a set of analog circuits to manage the read/write head and a set of digital circuits to communicate with the computer system. Although the combined specifications of all components determines the capacity of the disk, technological advances have resolved the issues to the read/write head and the disk surface. Disk heads are limited in their ability to "focus" a magnetic field on the surface of the disk and disk surfaces are limited in their ability to carry closely spaced areas of magnetized and unmagnetized (ones and zeroes) areas.

If we were limited to using the over-simplified configuration described above, our read/write head would be fixed in a single position over the disk surface some distance from the center. From this position, the head would be able to write on the disk in a circular pattern as the disk surface rotates beneath it. This pattern is referred to as a track. When writing to the disk, the head requires a small amount of distance to "stop writing" so that it will not overwrite good data. Also, when writing to a disk track, the write head actually will rewrite the entire track to avoid damage to good data.

To reduce the amount of time spent rewriting data, disk surfaces are divided into sectors. Sectors contain a header area identifying the sector; an area of dead space used by the head system to "start writing"; a data area (256 bytes for an HP 3000 disk); a dead area for the heads to "stop writing"; and a trailer

TABLE I




area to hold error correction data. By "formatting" a disk in this manner, we trade disk space for performance and data integrity.

Formatting also provides a simple method for compensating for surface defects. In the unformatted track described above, a single error would ruin the entire track. In the formatted track, only the few bad sectors would be lost.

Obviously, single-track, single-head disks aren't practical devices except in the trivial case of the magnetic tape drive. Early "fixed head" disk systems would place a series of heads at fixed locations along the radius of the disk surface.

Another approach to increasing storage capacity is to provide a mechanism that will move the head to and from the center of the disk surface radially. Each of the concentric circles thus formed will result in an additional track being added. Each of these tracks then is formatted into sectors to form a more useful storage device. Again, the read/write head and disk surface technology govern capacity, but now we can increase disk diameter to increase capacity also. Diameter does effect data acquisition speed, but we'll save that for later.

The next level of enhancement for disk capacity actually is a cost and space control measure that stacks several disks on a single drive mechanism and shares some of the electronics and packaging of the single surface system. The read/write heads generally are combined on a single control mechanism as well. In this configuration, the set of tracks formed by the heads as the disks rotate beneath them form a cylinder.

In multi-surface disk systems, tracks and cylinders are sometimes used interchangeably. The two terms are synonymous only when discussing head position. As disks are stacked, the weight of the head assembly begins to contribute significantly to total disk capacity, to the degree that it becomes difficult to accurately position the head quickly; in other words, it affects the ability of the drive system to accelerate

FIGURE 1

```
>pfspace 14;addr
LDEV:14
 ADDRESS     SIZE          ADDRESS      SIZE        ADDRESS      SIZE
 -------     ----          -------      ----        -------      ----

     5166          5         6766          20         7038          53
    29291         70        29592          19       191157         114
   191425         46       219868           4       250168          44
   250436         60       256470          63       256764          32
   263190         96       263517          69       281301          20
   434791        220       483839         122       505381         193
   674544         50       733344         208       873812         230
   888692        242      1168454          50      1265974          30
  1470084      42790      1514254       65662
 NO. ENTRIES:    26
 TOTAL VOLUME CAPACITY:    1579916 SECTORS
 TOTAL FREE SPACE AVAILABLE: 110512 SPACE
 MAXIMUM CONTIGUOUS AREA: 65662 SECTORS
```

FIGURE 2

```
:VINIT
VINIT G.02.B0 (C) HEWLETT-PACKARD CO., 1978
>pfspace 14
VOLUME   MUSIC00               LDEV 14
LARGEST FREE AREA= 65662
  SIZE   COUNT   SPACE    AVERAGE
>100000 0        0        0
>10000  2        108452   54226
>1000   0        0        0
>100    7        1329     189
>10     15       722      48
>1      2        9        4
TOTAL FREE SPACE=110512
***********************************
```

and decelerate the head carriage.

All of this technical background is only indirectly related to the management of disk space. It is provided as a backdrop for those who have had an otherwise great day ruined by a catastrophic disk error that requires disk formatting and initialization before a full system reload is done to get the system operational again.

It also points out that one should look into the formatted capacity of a disk subsystem, not just the unformatted capacity, because there can be a significant difference between the two. And finally, this look into the drive mechanism might spark an interest at looking into the mechanical specifications of the drives that are available on or for your systems. Note that more complex drive mechanisms are giving way to precision disk surfaces and head designs in the fight to increase disk capacity.

Let's assume that we have a drive of say 600 MB unformatted capacity. Due to sector size, head count and formatting requirements, the formatting process yields 404 MB of usable space. Let's look at how MPE works with that space before letting us users at it.

Again, we'll start out with a simple case — that of a system with only one disk drive. MPE calls the disk that contains system control information "SYSDISC." When a system is first started, the SYSDISC must be formatted and initialized by using an offline program. As the system is loaded onto the disk from magnetic tape, many data areas are loaded and reserved for use by MPE.

The first sector of the disk is loaded with some identifying information called a Volume Label. This is followed by a table of the bad spots on the disk surface called the Defective Sectors Table. Next comes a short program that's used to control the disk while the rest of the system is being loaded from tape, and a short program that's used to dump main memory to tape for diagnostics.

Next comes the program used to load the rest of the system and several sets of operating system table pointers that it will use. The accounting structure and file system directory is laid in next, followed by virtual memory, several system files and a table of free sectors on the remainder of the disk. (*Table 1* describes a typical disk layout.)

On a large (85-user Series 70) system, this overhead could consume 20 to 30 kilosectors with total overhead for all disks (six in the Series 70) amounting to 76 kilosectors. Once the system loading and control structures are set up, the remainder of the disk is available for "user files." Well, almost all of the rest. Actually, what we find is that, as far as MPE is concerned, most of the operating system and all software subsystems (EDITOR, FCOPY, COMMUNICATIONS, etc.) are "user files." Again, on our large system, this can amount to several hundred thousand sectors being consumed before any real user data and program files are accounted for.

In addition to the space required for MPE and system support utilities, space must be "reserved" for print and job spooling. Whenever a listing is sent to a spooled printer, it first will be written to a disk file where it will be held until the requested printer is available. Spooled print files are written only to certain disks — those that have been assigned a "device class name" of

# LABELS/3000+

**THE MAIL LIST & LABEL SYSTEM FOR HP-3000 USERS**

## •A comprehensive, user friendly, menu-driven system to provide:

- Mail Labels
- Envelopes
- Postcards
- Rolodex* Cards
- Specialty Labels
- Shipping Labels
- Air Bills of Lading (1)
- Index Cards
- File Folder Labels

- Complete on-line information
- Quick "Toggle" through records
- ★ Duplicate "Trap" at time of entry
- Probable duplicate indentification and deletion
- Find records by Record Number, Name (or part thereof), or Company (or part thereof)
- User defined Select and Sort criteria (including GT, LT, GTE, LTE, IB)
- ★ Stores Telex, multiple telephone numbers, and comments
- ★ Handles domestic and international address formats
- ★ User override of default label sizes
- ★ NEW LABELS/3000+ enhancement for those with multiple mailing lists. All mail list records go into one file while the system maintains the appearance of separate mail lists. No duplicate records for the same person! This could save a significant amount of disc space.
- Interface with existing files thru conversion/update programs
- ★ Sales Prospect Tracking module available.(1)
- ★ Unlimited category codes and user codes per record

~~$1800~~ **$1200 SPECIAL LIMITED OFFER**

(First CPU license - Object code only)
(Source code, modular pricing available)

**FREE 30 DAY - FULL FUNCTION DEMO**

## *WHY BOTHER TO DEVELOP ANYTHING LESS?*

FMT

**NEW PRODUCT: CAPTOOLS** - Three fast, efficient and flexible dataset capacity checking utilities that report current staus, previous status, and up to 99 previous times checked. On-line CAPCHECK is 10-12 times faster than Query FO sets. Object Code = $100. Object and Source Code = $200.

FMT Associates, Inc.
consultants & software developers
6475 fox run circle • jupiter, florida 33458
305-744-3652

★ NEW IN RELEASE 1.2 (1) OPTIONAL

* Rolodex is a trademark of the Rolodex Corporation

ENTER 147 ON READER CARD

SPOOL (another topic for a separate discussion). It suffices to say that spooled print files consume disk space on a transitory basis. When batch jobs are STREAMed, they, too, will be written to disk until an open processing slot is available. Consequently, WAITing and SCHEDuled batch jobs consume disk space as well.

Another "hidden" consumer of disk space is the TEMPorary file. Many user and system processes, especially SORT, consume disk space implicitly. Disk space must be made available for these processes to ensure against application failures. For completeness, I mention the *Passed* file. Certain system processes, and infrequently user processes, will employ one or both of two system-defined files called *Passed* files. These files are unique to each system user and are called $OLDPASS and $NEWPASS. Actually, the two files are one in the same with the name changed automatically by MPE from $NEWPASS to $OLDPASS when it's closed. *Passed* files take up very little space (usually) and are used almost exclusively during program compilations.

This brings us to the real user files. On most systems, user files will consume from 60 to 90 percent of the used disk space. Although techniques are available for managing the former categories, common sense dictates that user files should be the major target of the management program.

EACH HEWLETT-PACKARD disk drive is specified in formatted capacity. For example, an HP 7935 disk is stated to have a capacity of 404 MB. That would calculate to 1,578,125 sectors. Actually, the disk capacity is 404,458,596 bytes or 1,579,916 sectors. Heh! 1,791 sectors can make a difference on a bad day.

The program PVINIT.PUB.SYS has a command PFSPACE n;ADDR where "n" is the logical device number of the desired disk drive. After a long and painful display of every available disk region on the drive, the disk capacity will be displayed as shown in *Figure 1*. Luckily, the capacity remains the same for all disks of the same model number.

From the disk capacity, we must subtract defective sectors. PVINIT has a command PDTRACK that will display this information for some disk types, as will the SYSDUMP program. For other disk types, the program CS80UTIL.CS80.TELESUP must be used.

There's no direct method for deter-

FIGURE 3

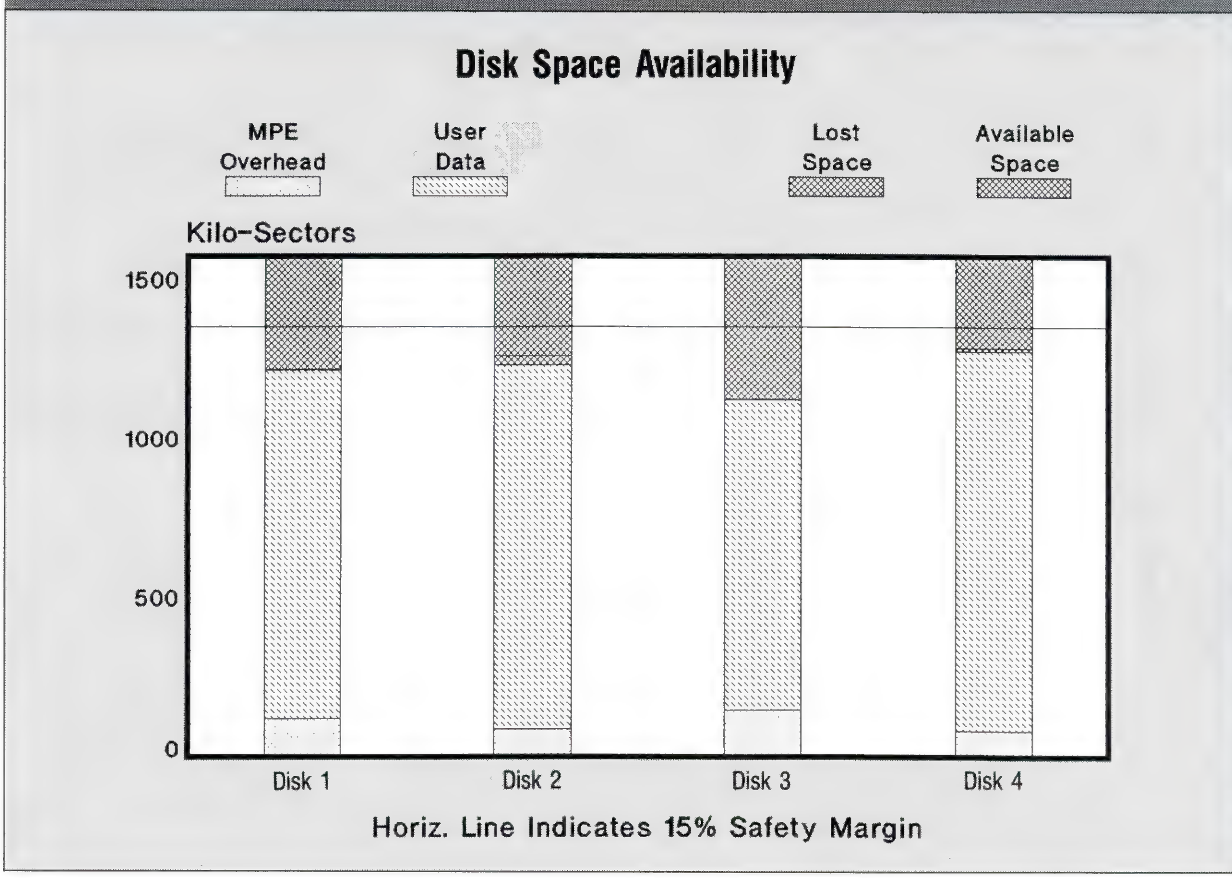


FIGURE 4

```
:RUN MPEX.PUB.VESOFT
MPEX/3000 Version 1.6 (VESOFT Inc. (C) 1980)  For Help type 'HELP'
%!LISTF @.@.@(DEVICE=1)
FILESET=  @.@.@(DEVICE=1)

ACCOUNT=  SYS         GROUP=  PUB

ADVWCATL    AM2563A     AM2686A     AM2934A     AMSPELL     AMWDBOOT
   :           :           :           :           :           :
WCSLE1      WCSLE2      WORDTEXC    WORDTEXT    X21MSG      XSORT

ACCOUNT=  VESOFT      GROUP=  PUB

DATECONV    GOD         LISTDIR5    MORTAL4     MPEXHOOK    MPEXINT
MPEXUDC     PROCS       SL          STREAMX     TELLX       VEMODIFY

ACCOUNT=  VOLZ        GROUP=  BASES

COMPUDC     EQSCHEMA    RPTMGT01    RPTMGT03    RPTMGT05    TAPMGT
TAPMGT01    TAPMGT02    TAPMGT03    TAPMGT04    TAPMGT05    TAPMGT06
TAPMGT07    TAPMGT08    TAPMGT09    TAPMGT10    USERDB01

ACCOUNT=  VOLZ        GROUP=  PUB

CATALOG     DIRSTATS    DISKHOGS    EQGET       EQMAIN      EQUATE
EQUDC       MANUAL      MPECMD      QMANAGER    QMGRP       QMGRS
RL          SLEEPU      TESTJOB     TV          VOLZ

GRAND TOTAL: 855 FILES, 189491 SECTORS.
```

mining the amount of space that is being used by MPE tables and control structures; however, there is an indirect method that will give a close estimate.

At the next RELOAD that you do on your system, use the ACCOUNTS option. When the system is up, RESTORE the program FREE5.PUB.SYS. A report of disk space availability at this time will give you a baseline. Subtracting the current FREE5 space from the available will give you your MPE overhead. (See *Figure 2* for a sample FREE5 listing).

The two major contributors to MPE overhead are the System Directory and Virtual Memory. The SYSDUMP utility will allow you to obtain your current allocations or to change them if desired.

The next step towards determining how much space is available to users will be to determine how much additional space is consumed by MPE and other Hewlett-Packard utilities. The SYS, SUPPORT, TELESUP, HPOFFICE, and HPPLnn accounts are all non-user consumers of disk space. Use the MPE REPORT (more on this below) command to determine the space consumed by each of these accounts.

SYSTEM LOGFILES GROW continuously as the system runs. If Predictive Support, HPTREND or any system accounting software is being used, the log files will grow very quickly.

It's important to remember that the logfiles must be on the system when the Predictive, Trend or accounting software runs, but they need not be on the system at all times. A Logfiles management program should provide for frequent archiving to tape after the periodic analysis is done.

HPDESK GLOBAL and LOCAL databases and associated files are stored in the HPOFFICE account. These files will grow continuously unless managed. HPDESKMANAGER IV allows limits to be placed on user workspaces. In addition, your HPDESK administrator should keep track of these files and work with the subscribers to limit online storage.

Databases generally will occupy a significant percentage of the user file space on any system. Database capacities aren't changed very often, but when they are, they tend to gulp down large amounts of space and keep it forever.

Your database administrator should keep track of database data set capacities and use to make sure that the files never get full. Master data sets should never exceed 80 percent of capacity and detail sets should be managed according to use. The primary issue with database files is that when full, programs will fail and databases can't be accessed while capacities are changed. A forced increase to a database file capacity almost certainly would result in significant application system down time. DBGENERAL from BRADMARK has a feature available that will monitor capacities and automatically increase or decrease them to your specification.

Although all files should be monitored to make sure that they won't overflow or waste space, there are few tools that will help with that task. A future installment of this series will deal directly with file management. Rather than spend a lot of manual effort on that task, there are other areas that will produce more valuable results.

Having determined the capacity and free space using VINIT as shown above, there are two management issues that must be addressed. Available space must be maintained and fragmentation must be minimized. Again, VINIT will be the measurement tool. The management program will:

- *Measure capacity.*
- *Establish minimum available space.*
- *Establish maximum fragmentation.*
- *Shrink or delete files.*
- *Add disk drives.*
- *Reduce fragmentation.*
- *Minimize lost disk space.*

It's important to keep track of disk space by disk drive rather than for the

system as a whole. Therefore, the best graphical representation will be to use sets of clustered vertical bar charts — one set of bars for each measurement period with one bar for each disk drive.

Each bar then would be constructed from four parts: MPE overhead in the base segment, user files in the next segment, available space in the next segment and lost space will complete the bar to reach capacity. (See *Figure 3*.)

MPE overhead doesn't change often, but should be recalculated after each system reload, each operating system update and whenever virtual memory is changed. MPE doesn't provide a method for determining the amount of user data on a particular disk drive; however, several utility programs will do the job. *MPEX* from VESOFT (Los Angeles, CA) can perform the calculation as can *DISKRPT* from Volz Associates (Winthrop, MA). (See *Figure 4* for a sample *MPEX* listing.) FREE5 or VINIT will report on available space. The last entry, lost space, must be derived from the measured data.

Temporary files, passed files and spooled print files will introduce errors into the data collection and calculation processes above; therefore, these measurements should be made only after clearing the machine of such or accounting for the current allocations in each category.

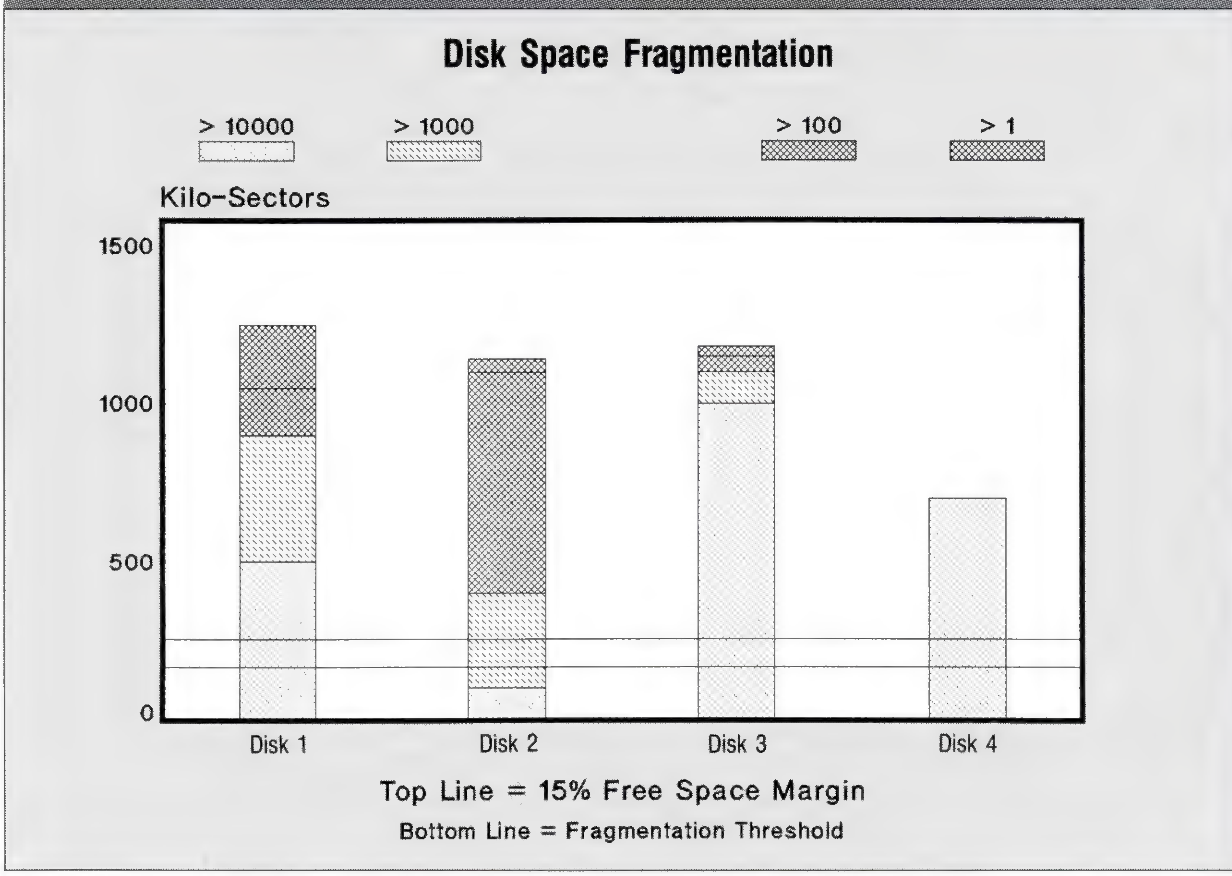
Available space doesn't imply usable space. As files are built and purged, fragments of disk space become scattered all over the system. If the fragments are too small, they'll never be used. FREE5 or VINIT will report on disk fragmentation.

The amount of fragmentation that can be allowed on a system depends on two primary applications. Every spooled print file will require a minimum amount of disk space to begin writing. That "extent" size is set through SYSDUMP and establishes the smallest fragment with which the spooler can work. The second major concern should be that MPE needs about 17,000 sectors to get started.

Fragmentation should be managed by setting a limit on the minimum amount of space that must exist in fragments greater than 1,000 sectors. Once that number is reached, one of two measures can be used to create larger fragments. (See *Figure 5* for an example of a fragmentation chart.)

A full system reload will create one very large fragment on each disk, but

FIGURE 5



this will take quite a bit of time and will take the system offline for the entire process. VINIT has a CONDENSE command that will create several large segments from many smaller ones, but this isn't effective at reclaiming very small regions. CONDENSE requires exclusive access to the machine and takes about 30 minutes for each disk drive. CONDENSE should be used frequently until the desired goal of 1,000+ sector fragments can't be reached, then RELOAD must be used.

Lost disk space was mentioned above, but never explained. There is no reliable way to measure lost disk space. We can calculate a value for it, but experience shows that the calculation is not repeatable with any degree of certainty. Lost disk space occurs when it's allocated from the free space tables, but is never returned when it's no longer used.

Several causes of lost disk space have been isolated at various times. If the system is cool started or cold loaded while spool files are outstanding, their space will be lost. Certain versions of MPE have been found to lose disk space when jobs or sessions log off with temporary files opened. Whatever the cause, measures must be taken to reclaim the lost sectors.

Both the cold load and cool start processes allow the "Recover lost diskspace" procedure to be executed. Recovery is done by accounting for all of the verifiable disk space utilization, then returning the rest of the allocated space back to the free space tables. The recovery process requires exclusive access to the machine and will take approximately 20 minutes for every 10,000 files that are on the machine.

Knowledgeable users might note that the MPE REPORT command wasn't generally recommended as a measurement tool. It would appear to be a useful way of determining how much space is allocated to user files. Unfortunately, in 11 years I've never known the data reported to be correct or even close for more than a few weeks after a RELOAD. I'll keep checking and hoping.

*As with any other management program, clear objectives must be set, goals for performance must be established . . .*

Some general housekeeping points to keep in mind:

1. Several types of logfiles should be attended to regularly. Consult with your account SE for details on where these come from and what to do with them:
   - *INPLOGnn.PUB.SYS*
   - *NMLOGnnn.PUB.SYS*
   - *LOGnnnn.PUB.SYS*
   - *IMAGE Logfiles*
   - *CSTRACE Logfiles*
   - *OPT Logfiles*
   - *APS Logfiles*

2. EDIT/3000 abandons so-called "K" files whenever EDITOR is aborted. These files will be named Knnnnn and might appear in any group of any account on your system. You can use the MPE STORE command with the ";PURGE" option to get rid of these.
3. People tend to leave everything on the machine forever. A good housekeeping operation is to use MPE STORE to back up all files that haven't been accessed in some time period (say, 18 months) to tape. Then repeat the process on another tape(set), this time using the ";PURGE" option to get rid of them.
4. Often programmers will leave print files on the machine for examination. Use SPOOK.PUB.SYS to back them up to tape and PURGE at the same time.

As with any other management program, clear objectives must be set, goals for performance must be established, measurement criteria and methods must be agreed upon, a monitoring program must be implemented, then corrective action can be planned and executed in advance of problem situations. There are no magic numbers for disk space availability, fragmentation or spooler space requirements. That data must be derived for each site and it should be expected that the data will change with time. Begin your program as soon as possible; monitor the disks and disk space-related problems; adjust your thresholds accordingly and take control of the disk space issue.

The next installment in this series will deal with batch production management and report distribution issues. Please feel free to send your comments and suggestions regarding system performance management to me c/o *HP PROFESSIONAL*, P.O. Box 445, Spring House, PA 19477. — *James Dowling is manager of Computer Services for Bose Corporation, Framingham, MA, and technical director of Volz Associates, Inc., Winthrop, MA.*

Would you like to continue to see articles on this topic?
Enter On Reader Card
yes 814 no 815

**Volz Associates**
34 Undine Avenue
Winthrop, MA 02152
(617) 846-3837
**Enter 605 on reader card.**

**VESOFT**
1135 S. Beverly Drive
Los Angeles, CA 90035
(213) 937-6620
**Enter 606 on reader card.**

**LANGUAGES**



# Getting Down To BASICs

**Don Person**

It's quite apparent that HP expects the Vectra's AT compatibility and speed to bridge the computer technology of the late eighties, if not act as a stop-gap measure to provide some breathing space for its own forays out of the INTEL CPU jungle. The generation of HP 386 machines ready to dash from the starting gate ought to reinforce that position still further.

While other hardware measures (most notably the VIPER) are being put in place to allow Rocky Mountain BASIC to "play" in HP's version of the MS-DOS environment, this question is not too far out of line: What is supposed to happen to those of us who in the past saw our HP investment as more than a "software toaster" for running ready-made applications? How do we write software for our own use on the Vectra and its INTEL descendants?

I've been spoiled by HP's high-quality, high-level languages. A quick perusal of the price list shows that to get the MS BASIC "Gee Whiz" interpreter, you have to fork over 350 hard-earned bucks. To distribute executable code, you pay another $395. For this much money, you have a language that's not exceedingly well endowed for technical pursuits and is imperfectly fitted to the interfaces essential for technical programming of this computer.

Compare GW BASIC to any of the native BASICs that made HP famous with engineering-oriented micro programmers in the past, and you'll see why I went looking for alternatives. The newest from Microsoft is QUICKBASIC (QB).

Other than the minimal search and replace, cut and paste and single line UNDO, almost any other editor is superior. In fact, these are the only features supported. It's an improvement over BASICA and ZBASIC, but decidedly inferior to what you get with either True Basic (TB) or Better Basic (BB).

While BB and TB can be used reasonably for program development, QB is not much of a contender. Assembly language seems indicated if you want to do more with it, though I think it's no accident that only two pages of extremely limited information on this vital gateway are provided.

It's not up to the standards for BASIC that HP has worked so hard to establish in the past and, in addition, GW provides poor access to the inner workings of DOS and BIOS interrupts that are part of Vectra's real strength.

If all I could offer was bad news about Vectra programming, we could quit right here. Fortunately, things are much better than you'd guess, once you examine some language options that don't happen to grace the pages of the HP price list. There are very good choices for do-it-yourself programming and complete application design.

PAST EXPERIENCE WITH GW-BASIC demonstrates that if you need a memory model that exceeds 64K, you have to look elsewhere. Getting access to system interrupts is no picnic either.

That leaves assembly language and C as viable options. Since neither particularly excites me, I felt that before climbing Mt. ASSM, I ought to look at other high-level tools. Without any fanfare, I have to declare two of the nicest Vectra languages for the casual programmer to be TB and BB. Feature for feature, these two are as remarkably similar to each other as they are radically different from GW BASIC. In an individual case, the balance is most likely going to be tipped simply by your access to specialized libraries or an expert tutor.

What I'll try to do is show you how they compare on the fine points of program creation, programmer productivity and ordinary ease of use. The views I have are based on the underlying assumption that you have some awareness of HP's Rocky Mountain or Series 80 BASIC language. For those of you with a relatively clean slate as far as programming notions are concerned, there are good reasons for you not to invest too much energy in GW BASIC.

Modern BASIC descended from FORTRAN, though perhaps fallen is the correct term, by way of the gutter. Good languages are not interpreted, they are compiled and executed in either a tokenized format or "in line." One of the hallmarks of the "new basics" is structural enforcement. If you're familiar with old basic GOSUBs, you'd better sit down. Each of these languages supports a powerful methodology that could change the way you view programming. Subroutines may be isolated from the main program, and can be reused far more flexibly than the old fashioned BASIC GOSUB you probably use now.

Another goody is the MODULE, which for the BASIC programmer is like a subprogram with retained data values and a host of other improvements. Subroutines are an expanded, infinitely more flexible version of what you may recognize as the subprogram in HP parlance. Now, write code once, and add

it to a personal library of infinitely reusable modules that are yours to command over and over again in other programs. This means you have progressively less and less new work to do each time you start a new task.

Also subject to large scale improvements is the User Definable function. The change is amazing compared to HP BASIC, and truly awesome next to GW.

Now a 13-round comparison of the top contenders.

## ROUND 1: Program Editing

TB 2.0 has a decidedly outstanding editor that's carefully tailored to the job of creating and rearranging large files with tools for trial as well as limited search and replace, and fast routine-to-routine scrolling through your file, great macros and an honest context-sensitive HELP to mention a few of its superb features.

BB uses an incremental compiler and requires line numbers, while TB allows them optionally, if it will make you feel any more at home. You use line numbers with TB only if you wish to use the old fashioned GOSUB and GOTO. In its non-numbered form, these are outlawed.

In retrospect, BB's insistence on line numbers is rather odd, and is made downright ludicrous by the fact that procedures and functions all begin with a list of non-numbered declarations! Changing one of these explicit (and occasionally implicit) lines of code requires moving heaven and earth when you see the dreaded "already defined" error.

TB uses plain string files for program storage, and while BB can import from this format, it can be agonizingly slow to make the conversion.

Because type declarations are binding in BB, and the text of procedure code normally is hidden during list, this round has to go to TB. This decision is reinforced by the difficulty of "backing into" BB code changes, particularly as it effects the necessity to predeclare variable types. Version 2.0 of BB has big improvements in variable handling, but it's not up to the level attained by even HP 80 BASIC's designers when they wisely chose to sever allocation from editing.

Changing the number or type of arguments in BB can be quite troublesome, while easy as pie in TB. TB allows infinite second chances in editing. Made a boo-boo? Just recompile after the change. Blew it with BB? You may have to stay late to undo a careless reference to a variable that you forgot to explicitly declare.

The editor provides little help if you need to change a type declaration. While both editors allow global or local changes, TB's is certainly easier to visualize. No part of the file is ever hidden, as is the case with BB. BB allows 255 characters per line. TB 2.0 can han-

## [ So What About ZBASIC? ]

ZBASIC is in many ways a functionally disimproved BASICA work alike, with a few important differences. It has procedural isolation and self compiles, but doesn't hold a candle to BB or TB. Yes, it has full 640K memory support, but it only handles strings up to 255 bytes.

A more critical failing is the editor. It was fathered by the "one line at a time" school of thought. With BB and TB you roam the screen and if you can see it, you can move to it and edit. I knew I was in trouble with ZBASIC when I couldn't find reference to the up and down arrows. Yup, they don't do anything. To edit a line, you have to ask for it to be taken out of cold storage. Then it's sub-Edlin all the way.

I can say on the plus side that it is:

- *Compact.*
- *Decently documented.*
- *Produces the smallest EXE files of the three.*
- *Has a nifty internal sorting keyword.*

But the minuses are hefty:

- *Short strings (255 bytes).*
- *No facilities for calling interrupts.*
- *A complete lack of user-defined functions and procedures.*
- *No programmers aids to speak off.*

In summary, it works, and if it had a real full-screen editor, instead of an old fashioned substitute, it might be OK. It's a good language for plinking around, but minus #3 knocks it out in the first round.

dle 65,532, though I don't know how or why you'd do this with only one statement legal per line.

Another desirable aspect of TB is a first-rate keyboard reassignment scheme with macro features that can be stored in a file for instant recall. A real nice feature.

TB even has a mechanism to allow large literals to be moved between the command/history window and the file edit window.

BB's editor lives in the same window you work in, and gives instant access to any modules you had linked when the language was booted. (TB allows compiled code and modules to be loaded into your workspace.) If you've used HP BASIC in the past, you will be at home instantly with BB's editor right down to the "feel" and instant syntax check.

This is not to say that there are no loose edges. BB has some troubles with forward references to non-existent functions and statements of which TB is inherently free, but this points to the underlying assumption in TB that it's meant really to do its stuff in runtime only. This may not be as useful to the part-time programmer as BB's slick and responsive immediate mode.

TB has just one more worthy feature that I didn't immediately catch on to. It's the DO COMMAND, not to be confused with the DO/LOOP structure. A DO program is special in that you can call it to work while another program is being edited — sort of a real-time program code filter, with properties that can be applied to the entire operating system. It's stored like any other declared EXTERNAL (Library) module, with the exception that the first subroutine in the file should have two arguments. One is a string array, and the other is a simple string.

Both TB and BB support a "pass through" to the DOS command interpreter shell. Also very neat.

A parting shot on installation: BB needs to access a special file called BB.CNF which *must* be in your root directory. This makes hard disk operation a bit more of a hassle than I like, particularly when I want to organize my different languages in their own subdirectory. BB makes you run a long install program with menu after menu of questions about your system. It's odd, since much of what is asked could be found out by the program, even by the language itself on loading.

TB has no configuration process at all. It finds out if you have graphics, color, EGA-installed memory, etc. It's all automatic each time you load up. This is the way things are supposed to work. More puzzling is that TB makes no claim of exclusivity to the MS-DOS world. They actively are making versions go on the APPLE MAC and AMIGA to name two other systems. BB is only for MS-DOS and the PC, and still you have to do its homework for it.

TB does have an optional config file that can be used to preload your favorite library modules, or set up your favorite keyboard macros, even load your last workfile if you like, all automatically.

TB program files always are stored as collections of ordinary ASCII strings with .TRU as an extension. Don't like the TB editor? Use your own.

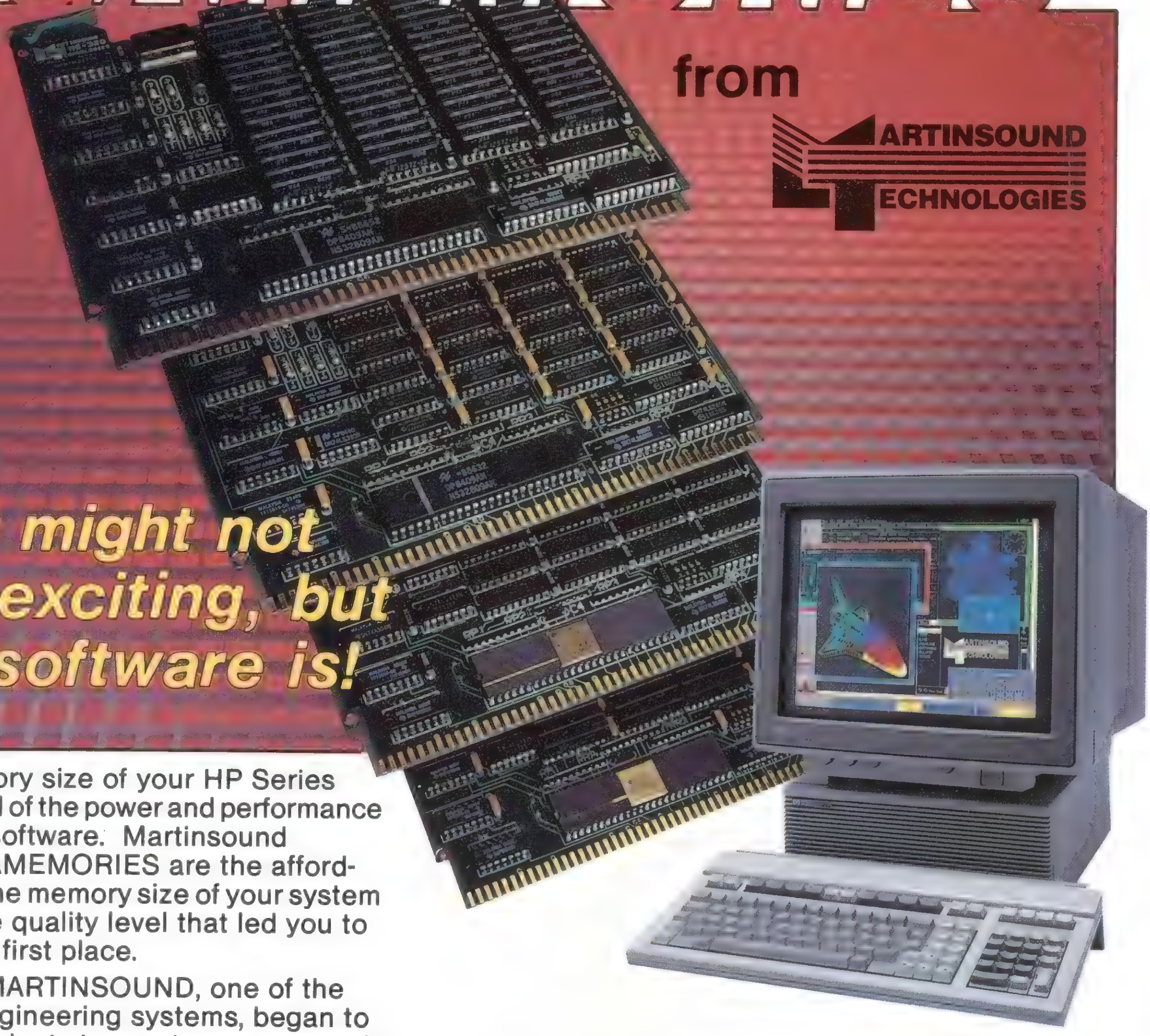
### ROUND 2: Structural Enforcement

TB does no structural checking during editing. BB prohibits GOTO in or out of blocks, but will allow branches internal to the unit. Both have good implementations of the DO loop with WHILE, UNTIL and other constructs legal at

# MEGAMEMORIES™

from

MARTINSOUND TECHNOLOGIES



*Memory might not be very exciting, but today's software is!*

Expanding the memory size of your HP Series 200/300, gives you all of the power and performance available in today's software. Martinsound Technologies' MEGAMEMORIES are the affordable way to expand the memory size of your system and still maintain the quality level that led you to an HP system in the first place.

Over 16 years ago MARTINSOUND, one of the early users of HP engineering systems, began to design and build products to meet our own need for high-technology products. We are not an importer or reseller of off-brand or off-shore clones, we use what we build. . . It's our efficient all domestic design, manufacturing and marketing which allow MARTINSOUND to offer the MEGAMEMORIES at prices which are as much as 60% less than comparable HP products.

A size to fit any application:

| | |
|---|---|
| **MS-310**<br>**1 MEGABYTE** | The affordable alternative to the HP 1Mb board. |
| **MS-320**<br>**2 MEGABYTES** | Twice the memory, for less than the cost of an HP 1Mb board! |
| **MS-340**<br>**4 MEGABYTES** | The industries best Mb-per-dollar value, and you save the cost of an expansion box. |
| **MS-380**<br>**8 MEGABYTES** | The ultimate slot-saver! In-stock now, not just a promise for the future. |
| **MS-350+** | MEGAMEMORY upgrade for HP 350 users now available! |
| **RAMDISK** | 1Mb Battery backed-up high-performance RAMDISK. |

MARTINSOUND backs what it builds with a full two year warranty, 90-day money-back guarantee, and 24 hour service, if ever needed.

Whether you are a corporate, government, educational or end user, MARTINSOUND has always had the lowest price available - even in today's volatile IC market. Call us before you buy anywhere else and check it out, you owe it to yourself.



See your local Martinsound Technologies dealer, or

**CALL US TODAY TO PLACE YOUR ORDER!**

**(818) 281-3555**

*"American Built Quality and Service Since 1971"*

MARTINSOUND

Martinsound Technologies
1151 West Valley Boulevard, Alhambra, CA 91803-2493
Telex: 85-6573

ENTER 115 ON READER CARD

both ends of the block. FOR NEXT is there, of course, as is IF THEN ELSE / ELSEIF, but TB adds to this a great implementation of the SELECT CASE statement with a nice range of possible arguments, including branches on:

- *A range of conditions*
- *Complex expressions*
- *String expressions*

The statement has its problems when a range of choices needs to be evaluated by both type and precedence. It's not much good for problems of this sort, while BB, on the other hand, supports the computed ON GOTO/GOSUB which is! TB has no similar feature for branching. SELECT CASE also can be notably inefficient in the time domain. This is an area where TB may be weak as a whole, since in a number of areas the developers have succumbed to the tantalizing lure of using the language to define itself. The result is families of standard features built upon other modules. While superficially alluring, the sluggishness of certain high-level modules is the price you pay as a user.

A terrific feature supported by both BB and TB, and already familiar to PASCAL programmers, is the independent module. A module is a code unit that has independent, but preserved internal variables either hidden or made public to any program that uses it. It has its own private initialization routines, and pass lists for shared variables as well. This makes global variables possible and can do a lot to shrink pass parameter lists too, as long as it's done with care. You have to use it to really appreciate the added power and flexibility.

*A terrific feature supported by both BB and TB, and already familiar to PASCAL programmers, is the independent module.*

## ROUND 3: Variables

There are soft spots in TB, and this is one. All strings are dynamic. That is, you never need to declare strings. Then again, you have no control over string positioning in absolute address space. Worse still, a string address can change, even if you have made no reference to it. In TB, you do have to declare the number of array dimensions and upper bound, but this can be changed at will within a running program. TB cleverly obsoletes the OPTION BASE concept.

BB supports a range of variable types and is much more flexible. Dynamic strings can raise hell with a program that uses an assembly language interface, because there's no assurance that a string pointer that was valid when you set it will be good when DOS uses it, particularly if you're using a string for a DOS table pointer with an interrupt vector changed to point to it, or need to worry about DMA across a 64K boundary.

BB allows dynamic strings, but also can handle a dimension declaration that fixes the size of a string, and its absolute address for the duration of the program. On the down side, it does no garbage collection for dynamics, so you may run out of RAM if you don't understand the strategy required. This is easier than you think, since BB doesn't explain it at all. BB dimmed variables do stay put, however. This is important if you need an assembly language interface to the operating system.

BB also has the best range of choices in variable type. TB forces all numerics to be REAL, with attendant foolishness and wasted time if all you need is an integer- or byte-sized counter, for instance. This can take its toll in multiple-nested loops. In spite of this enforced uniformity, TB scores high in the looping test.

Another advantage is that BB supports pointer variables. You declare a pointer and set it to point to another variable. Whatever you do to the pointer is done to the variable it points to.

Many other slick variable manipulations are provided, which can help you deal with BB's deficiency in explicit substring management.

TB has complete and well thought out support for substrings. Though BB improves in many ways on GW BASIC, it still adheres to the moronic LEFT, RIGHT, MID baloney used in BASIC. TB supports true substrings much like HP BASIC, by direct reference, and is easy to adjust to.

Here is a factor that completely could tip the scales in favor of TB. Do you do much in the way of matrix math? TB has a significant number of keywords devoted to MAT statements, all in forms that will be pretty familiar to HP BASIC lovers. Array bounds also can cover any range you define.

There's also feedback on upper and lower dynamic bounds of an array, since these can be changed at will in runtime. Increasing the upper bound of a one-dimensional array on the fly doesn't destroy TB variable contents either. BB has none of these extensions.

BB 1.0 supported BCD math with REALs to .1E-245 to 9.999 E+253 with 24 digits represented. Integer is +32768 to -32767. Byte values are 0-255. Version 2.0 now follows the IEEE math recommendations.

TB uses IEEE math, too, and has automatic support internal for the 8087/80287 coprocessor. I did all my tests without the 80287 coprocessor.

## ROUND 4: Functions

Both contenders have plenty of functions, whether predefined or placed in external modules. Since you can cook up

more of your own, it would seem the vista is limitless. But look closely. There are some differences.

TB is inferior in access, if not in approach as well. Many important statements in TB really ought to have been written as functions in order to take advantage of multiple assignment and to simplify coding, particularly when a function's value is of purely transient interest. For example, to find the row and column coordinates of the CRT cursor, you do this in TB:

```
ASK CURSOR row_var, col_var
```

The two variables receive the information. This should have been implemented via predefined functions such as:

```
ROW
COL
```

Key input and many other statements-cum-function could have been done better had it not been for blind insistence on making an ANSI standard language.

Functions that you define yourself are about the same in both BB and TB. Since there's no real internal organization for fast lookups, it's not uncommon if your TB functions are part of a larger program to have to wait 10 to 20 seconds before you see results in the history window. That's two interactions per minute. In short, TB is less quickly *non*-interactive.

BB, though it lets you use functions the instant you create them, makes it rather difficult to change them once you've started a session. It's the overly rigid variable typing/declaration fault I've already mentioned. This round is a draw. TB can grant slow function access to the editor, while BB is overly rigid.

In TB, user functions may be single line, or as many lines as you'd like. There is no practical limit to the number of arguments or their type. String functions may return up to 65,533 characters, while BB is limited to 32,766. This is also the maximum length of a string or an element in a string array.

Functions included in libraries can be loaded at any time during a TB session, and become accessible immediately for your use in the command window. BB's version is always loaded before an edit session, and is also at your disposal while editing or running programs. TB lets you clear out the environment, too. It's as laudable as it is useful.

On the other hand, BB requires that the new function be written, debugged and compiled before you even can enter it into a program. It's much more difficult to root out references to a faulty function once used with BB, and I see this as potentially quite problematic. With TB, if a function needs revision, you simply recompile just the faulty module, and the problem is solved.

## ROUND 5: External Subroutines

This is where BB outshines TB. BB subroutines virtually are user-definable statements. Arguments passed to TB external subroutines (or procedures, as BB calls them) must match exactly in type and number or an error is called. BB has a great feature here, in that it allows you to declare:

- *Optional arguments.*
- *Undefined types.*
- *Specified default values for each of the above.*
- *Families of procedures with the same keyword, but unique syntax.*

The system figures out which you mean to use based on the argument list.

In a word, absolutely superb.

In case you didn't get it already, subroutines (procedures) are like HP subprograms with a complete life of their own, including local variables and a pass list. But like true statements, they can be defined to stand alone, without parentheses enclosing the arguments. Pass by argument or value. What's your pleasure?

Both BB and TB have mechanisms for setting up global variables. In a large program, this can save you a lot of wasted time in passing variables back and forth. Localization of values can be a problem if you don't watch your step with either BB or TB.

Finally, a performance complaint directed at both languages equally: While subroutines improve readability and modularity of programs, I've found a dark side, too. Execution speed can suffer from fragmentation. The overhead of getting in and out of nested modules can degrade runtime performance. I found myself crafting individual modules that made good sense, only to merge them with part of a larger structure in order to get back the speed I was losing to the "organization" I'd achieved. A false economy to be sure. I believe now that structure is in the mind of the beholder.

BB does seem to be somewhat bound by procedures, but TB visibly has committed what I consider to be a major sin in its use of the language to define itself. The bit-level subroutines in TB are a minor disaster, in that they're built on the shoulders of other TBs, and execute with pitiful slowness compared to the same stuff in BB. The AND, OR and XOR routines in the developers' toolkit are a shameful example of coding that can be done in TB, but would have benefited us all had it been done with assembly language instead.

TB's other structural omission is the NOT operator. Boolean operations didn't seem to be very important to the TB people. Here's a statement that's legal in BB, but not in TB. Points subtracted from TB's score.

```
IF A+B THEN LET B=0 (Causes
a compiler error in TB.)
```

When we examine the requirements for a do-it-yourself assembly language interface, TB does give a much better explanation of how to do it. This ultimate extension is not well documented in BB, and was quite puzzling initially.

## ROUND 6: Disk Access

Pretty much a draw. BB and TB each makes a good interface to DOS capabili-

ties for disk and file management.

BB has far more in the way of random record blocking, and matches MS/BASIC conventions. TB may be stronger here in that it follows HP BASIC for random records much more closely than BB. Take your pick. Each has good points, each some bad. I've never liked the field formatting nonsense that GW BASIC requires for random records, but BB's pointer system makes it a bit easier to swallow.

**ROUND 7: SYSTEM/DOS Routine Access**

How does the dialect let you call soft interrupts? Since ordinary MS BASICS won't let you have this direct access to DOS, here's a strong endorsement to use either TB or BB to do things that aren't possible otherwise. Since each lets you work with 10 times the memory of MS BASIC, there's more reason to cheer (640K vs. 64K).

The more familiar access to the system is with PEEK and POKE. Here, BB is the loser. It clings stubbornly to the segment and offset concept of the INTEL CPU itself, originally promulgated by Microsoft. BASIC.TB lets you treat all memory as a single flat address space: BB uses the DEF SEG / OFFSET junk of MS BASIC, and requires that you define a 64K block for PEEK and POKE before using it.

An anomaly never explained is why the BB "SEG" of a string function always returns a negative number. This is maddening when trying for access to substrings. Me thinks it might just be Mr. McBug protruding, given several strange crashes when trying to PEEK and POKE substrings directly.

Of the two, TB has better provisions for calling interrupts. You can have easy access to any bit of any of the major registers. TB uses "bit packing" to put numbers into string form, and thence into a register string that loads CPU registers before an interrupt, and passes the resultant values back afterward.

BB has coarser control in the form of still more defined keywords for register PAIRS. To access the AH register alone, you must still assign to the AX keyword before calling the interrupt. Like TB, the carry flag is passed back for error status reporting.

To make your interactions smoother, TB's runtime library has a function called string__ptr which gives the equivalent of a VARPTR. It's simple to use once you realize that STRINGPT (as it is called) is aimed at the HEADER of the string, eight bytes ahead of the first actual character.

Both support passing commands through to the DOS command interpreter within the language. TB 2.0 goes further in that you may run another application from within TB and then resume where you left off. A very good feature if you have the memory.

**ROUND 8: DEBUG Facilities**

This round shows BB's better side. When you have about a thousand lines of code in a TB program (easier than you think with only one statement per line), asking for a variable value during a breakpoint or after execution is exasperating.

You can wait for over 30 seconds for the system to respond when you ask a question. Debugging TB can be a drag. Breakpoints and tracing are both available, but variable inspection is awfully tedious. Need I elaborate further? BB by comparison is lightning quick.

TB has another nasty habit. At a breakpoint, when I ask for a variable's value, I frequently see the message: "ERROR 777 caused." Naturally, this error number is not listed in the manual. There are a few other cryptic, messageless errors, usually called because you loaded some dud code into the workspace with the load command. This round is BB's, uncontested.

**ROUND 9: I/O And Hi Level (or "soft") Interrupts**

Programmable interrupt capability is taken for granted in most HP BASICS. When you use an ON TIMER, ON TIMEOUT, PAUSE or the ON INTR statement, you're using the end-of-line interrupt. Only BB can do this. TB has no provision at all, and the written information suggests that you poll modules with software of your own creation "in a timely fashion" to see if service is required!

If you absolutely can't do without I/O interrupts, BB is the only choice. This is another of those real world requirements that seem to have fallen through the cracks at TB. TB does have a nice serial support module available separately as the "Communication Support" option, but this round is BB's.

**ROUND 10: CRT And Keyboard Control**

TB has some very good handles on the CRT. There is multiple virtual windowing and X-Y position feedback. Any combination of graphics and color is recognized automatically without the

*TB has some very good handles on the CRT. There is multiple virtual windowing and X-Y position feedback.*

**Better BASIC**
Summit Technology Inc.
106 Access Road
Norwood, MA 02062
(617) 769-7966
**Enter 600 on reader card.**

**QUICKBASIC**
Microsoft, Inc.
10700 Northup Way
Box 97200
Bellevue, WA 98009
(800) 426-9400
**Enter 601 on reader card.**

**True BASIC**
Addison Wesley Educational Media Systems
Jacob Way
Reading, MA 01867
(617) 944-3700
**Enter 602 on reader card.**

**ZBASIC**
Zedcor Inc.
4500 E. Speedway #93
Tucson, AZ 85711
(602) 795-3996
**Enter 603 on reader card.**

need to write a CONFIG file, as you do for BB.

TB does not wrap character output modulo the window size. This can make for atrocious-looking printed output within a window. BB makes windows, too, but preserves the "underlay" for reversals so that windows can use a space temporarily without losing what was underneath.

BB has a built-in fast CRT alpha character read statement ala HP BASIC AREAD, called CHAR$ too. To do this with TB, I had to write my own DOS interrupt call. For ALPHA output, BB is my choice. TB uses only reliable, but sloooow DOS calls. BB interfaces directly to video RAM by the feel of it. It's great for now, but changes in video mapping for new HP products might leave it in the lurch later.

This can be a major consideration when developing your own programs. I believe that even though it is substantially slower to use DOS-provided CRT access methods, it's the only way to go if you wish to insure that your program runs on any of the near compatibles, and especially 80386-based systems. BB puts a slick look on the CRT, but may be playing a bit too fast and loose with this essential interface. I award this one to TB, in spite of its slight infirmity.

## ROUND 11: Graphics Support

Now it's TB's turn. While BB follows the GW conventions for graphic output (more or less), TB has added more elaborate drawing and graphic instructions. While neither of the pair is particularly close to the outstanding level of plotter/vector drawing commands found in HP basics, TB will be closer to satisfactory. TB is the winner by a nose, but I sense the market for third-party modules with SUB calls that exactly mimic HP BASIC plotter statement syntax, translating to HP-GL via the serial interface. There might be a market for HP-specific support modules, too. Hmmmm.

## ROUND 12: Producing Executable Programs

Under MS-DOS, the executable file is a necessity for commercial programs, since you don't want to have to distribute copies of a separate BASIC environment. This is true also for home-made code that you'll use frequently yourself. Even when starting with compiled TB programs, there can be an appreciable wait for pre-run linking. TB and BB have runtime programs that can make your code free-standing. Of the two, TB is easiest to use. You compile your modules and put them on disk. Then call the TB binder. When it's done, you have a packaged program ready to call at the DOS prompt.

GW/MS BASIC compiler can't hold a candle to this. BB requires more fiddling around to get to the same place as TB, but is still far superior to the MS compiler, with its plethora of flags, foolishness and frustration. TB gets the decision on its utter simplicity.

## ROUND 13: Documentation

There's a clear winner and loser in this round. BB's book is quite chubby, but most of its mass is devoted to a voluminous though slightly incomplete alphabetical syntax guide. It takes a serious fall to the canvas for its dramatic lack of examples. The examples and overview are far too brief for my liking, and there is effectively no information about internal organization.

The most difficult portion to comprehend is the random access file and pointer concept. The problem is that once you produce a working language, the job is only half done if you fail to clearly demonstrate it. This is what has happened with BB.

TB is another story. Perhaps owing to the Addison Wesley publishing connection, or the teaching direction chosen by the team, the pair of manuals that come with TB rate a solid good + on the scorecard. However, the overall impact is marred by some scattering of relevant information between the volumes for no apparent reason. Documentation for many option modules is presented less completely.

For learning, and the one trick pony programmer, TB is a good choice. The "pro" programmer who is not stymied by shaky documentation and quirks easily could use BB.

TB's clarity of documentation and general feature-for-feature parity so overshadows BB that I've decided to hang my hat with the long suffering team in Hanover. When you split your time between diverse languages, intuitiveness and the quality of structure and syntax information matter a lot. True BASIC meets nearly all my criteria for a great Vectra language. *—Don Person is an independent consultant based in Albany, New York.*

Would you like to continue to see articles on this topic?
Enter On Reader Card
yes 808 no 809

SPECTRUM

# What If . . . You Didn't Wait For Spectrum?

**Michael Shumko, Robert Green, David Greer**

The problem: *Nightly batch jobs are still running the next morning. Users are complaining that online response is terrible. In short, your HP 3000 is overworked, under paid and about to collapse from exhaustion.*

The solution: *Order a Spectrum — a Series 930 or 950.*

The problem with this solution: Neither machine will be in your shop for weeks, maybe months. What to do?

The solution to the solution: We've done an information survey of large HP shops to find out how the successful ones avoid topping out the HP 3000 line. What we found was not some "secret" formula, but rather a mundane, continuous attention to the details of system performance. The successful sites still apply the long-touted answers for boosting performance, such as balancing use of disk drives. Just look in your magazines, newspapers and conference proceedings for all sorts of ways to improve performance.

Here are some of the ideas these users mentioned for how to squeeze the last bit from your HP 3000:

- *Pay attention to details.*
- *Use one CPU per problem (distributed processing).*
- *Distribute an integrated solution over several CPUs.*
- *Put heavy CPU work on PCs (word processing, graphics).*
- *Upgrade to faster hardware (Series 70, LAN, forms cache).*
- *Review batch processing.*
- *Use NOBUF tools and optimum block sizes.*
- *Compile your fourth-generation applications.*
- *Get OMNIDEX for fast online database searching.*

Not all of these solutions will apply to everyone. Many are "old hat," but they work. A few of these ideas are novel — you may not have heard of them before. Some are not cheap (then again, neither is a Spectrum). If you're trapped for horsepower *now*, then these timely suggestions may give you the breathing room you need. Until Spectrum, of course.

## Attention To Detail

Good system managers never stop thinking of new ideas to improve system performance. Successful sites are constantly monitoring their machines. Here are some of the tools.

■ *Response Time.*
Most shops we surveyed had OPT (OPT, Inc., Upland, CA), a measurement tool that shows what is going on in the machine, but most were not using all of its features. David Lustig of Bose Corporation (Framingham, MA) uses a simple method to measure response time. When a user complains that the computer is slow, he goes to the user's terminal and uses a stop watch to time the actual response time.

Other sites were using SUE or SURVEYOR from the Contributed Library.

■ *HPTREND.*
Most sites are using HPTREND to provide accurate information about machine use. In many cases, the HPTREND reports confirm the system manager's intuition and provides concrete evidence for upper management. Sites we contacted are planning their machine resources at least a year into the future. Jim Bird at Turbo Resources (Calgary, Alberta, Canada) is trying SYSPLAN (Carolian Systems, Toronto, Ontario, Canada). This product is similar to HPTREND, but the trend analysis is done on your own machine.

■ *Database Performance.*
Turbo Resources uses HowMessy (Robelle Consulting) to identify the inefficient datasets in its application (HowMessy is run once a week). Turbo uses DETPACK from Adager (Guatemala) to repack one critical dataset every day. HowMessy was used to obtain the "before" and "after" pictures of the dataset packing.

■ *Disk Cache Optimizer.*
Markku Suni, an SE in Finland, has written an unsupported program (DCO) that manages disk caching parameters dynamically. It varies the sequential and random fetch quantums depending on the current job mix, I/O queue lengths, etc. It even will disable caching on a drive if it decides that throughput would increase without it. DCO doesn't work well when the machine load is extremely dynamic (e.g., on a development machine). You can obtain a copy from your SE.

These are a small sample of the ideas that help monitor and improve system performance. No one knows all of the details that will keep your machine running; you must strive to find them.

## Use One CPU Per Problem

Problem: *How do you add CPU power to a 3000 when you already have a Series 70?*
Solution: *Use one CPU per problem, or application or department.*

Don't try to crowd everything onto one computer. Instead, use a separate CPU for each major application, or give each department its own machine. That way you make each application independent of the problems in other appli-

*(List of vendors can be found on p. 98.)*

cations. If the payroll application is a hog, there's no reason for the accounting users to suffer.

Using separate machines also allows you to tune each machine for its own application. "Distributed processing" was the strategy most frequently mentioned in our survey of successful sites. Most give programmers their own machine.

At Boeing, one of the large manufacturing systems has an "update" machine and an "inquiry" machine. The update machine has 150 users who are updating the database. No uncontrolled inquiries or reports are allowed on this CPU. The inquiry CPU has a copy of last night's database from the update CPU; on this machine they allow people to make inquiries and to run QUIZ (Cognos Corporation, Peabody, MA).

HealthPlus of Michigan provides healthcare services using a Series 70 with 52 sessions for all data entry and a Series 68 with 30 sessions for all online inquiries and reports. They use silhouette to keep the inquiry database current and a Series 48 is reserved for all program development. Word processing is done on two standalone Series 37 machines.

Longs Drugs, a large West Coast chain of drug stores, has 200 HP 3000s. An extreme example? Not really. True to the distributed processing ideal, each store has its own Series 37. These handle the main pharmacy application, keeping track of prescription stock, filling orders and checking for dangerous drug interactions.

When required, the Series 37s use dial-up distributed systems (DS) to exchange information with the head office Series 70s. Otherwise, they're standalone machines. Every machine has a Console Engine from Telamon (Oakland, CA) to let the head office know when problems occur. (In fact, the Console Engine initially was developed for Longs Drugs.)

At the head office, Longs puts separate applications on separate machines. For instance, all the personnel applications are on one Series 70, the accounting applications are on another. Development is done on a separate machine.

Consider another example, a company that sells supplies. It has 18 HP 3000s spread all over the world. Before the MIS manager went to work there, his philosophy was always "get a bigger machine." Then he went there, where the company has a philosophy of "getting the data down to the users." So, they have 3000s everywhere; every warehouse has its own small HP 3000.

They were having a problem with FA/3000: They gave it its own Series 58. They don't even have a Series 70, and aren't budgeting for one until fiscal '88.

■ *Tools.*
If you take this route, you'll want to make sure you have the proper tools to manage the network machines properly. One type of tool is used to route spool files from one machine to another conveniently.

Unispool from Holland House (Beeville, TX) is one example of this. It allows you to have an expensive peripheral like a laser printer connected to one machine and have more than one computer send output to it.

RSPOOL, from the Contributed Library, will duplicate spool files across a DS line. RSPOOL creates a remote session, runs a remote program to generate the remote spool file, and purges the original spool file. The price is cheap, but Joe Ballman of Textron Marine (New Orleans, LA) warns that RSPOOL eats up the LAN and consumes over three percent of the CPU.

$Stdlist Management Software (now called Job Rescue) from NSD, Inc. (San Mateo, CA) also can help; it checks spool files for error messages. This lets the computer look for problems itself, allowing the users to get on with their own work instead of babysitting the computer.

*The key advantage that system managers see to the "one CPU per problem" philosophy, in addition to never "topping out," is that you can push the machines into the user environment.*

If you set up a machine in a user department for unattended operation, you'll still have console messages to contend with. The Console Engine attaches to the console and looks for specific conditions such as system failure messages, error conditions, and that sort of thing. If it sees that the system has run into some trouble, it either can take action on its own (a "pseudo operator") or it can dial the head office and notify the system manager.

■ *Resist Getting A Bigger Machine.*
You always can have that in reserve if you get into trouble. Get another machine instead. Dexter Shoes manages one million open items, one million inventory items, six show factories, over 50 retail outlets and numerous wholesale clients with a network of six Series 40s.

The key advantage that system managers see to the "one CPU per problem" philosophy, in addition to never "topping out," is that you can push the machines into the user environment. You don't have to have a giant MIS. And when the machine is slow, it's because

the users are running QUIZ reports. There are only a dozen users, so they can observe and figure it out, whereas on a Series 70 with 180 users, even the system manager doesn't know what's causing the problem. So, you break it into smaller problems. Each machine is less complicated and, we would guess, has fewer problems. You will pay a little more for maintenance and raw horsepower, but you should be repaid easily in better user service.

*You should not allow any word processing on your HP 3000 unless it's dedicated to word processing. HP has been advocating this approach for a few years.*

## Distribute An Integrated Solution Over Several CPUs

Okay, I accept the idea that I should have one application per CPU, but my application is an integrated solution. All of the modules access common databases and I don't have time to rewrite it (or I bought the package and I don't have the source code).

Problem: *You can't split a single integrated application over two machines.*

Solution: *Yes you can, if you're clever.*

■ *AutoNet.*

Karl Smith of Softsmith (Houston, TX) has developed an ingenious simple method of distributing an integrated application over several HP 3000s. Compaq Computers (Dallas, TX) started in business a few years ago. To manage their manufacturing work, the folks there used ASK Computer System's (Los Altos, CA) MANMAN system over dialup lines to a time-shared Series 42. Within weeks they had their own machine, then two, and so on. Their growth has been so dramatic that they've never had the time to customize ASK's programs — they use them "as is."

Compaq now runs its entire company on a network of 900 PCs and seven HP 3000s (no IBM mainframe). When their processing needs for MANMAN exceeded the power of a single Series 70, Tom Callaghan hired Karl to program a solution.

Tom wanted to be able to spread the databases and files of the integrated MANMAN application over more than one HP 3000. Karl wrote an SL routine to intercept all calls to the FOPEN intrinsic.

His routine, called Global FOPEN, checks the user's desired filename against a table of remote file-set names. If it doesn't find a match, Global FOPEN calls the real FOPEN. If it does find a table entry for the filename, Global FOPEN automatically gets the user a remote session with the same logon as his local session (unless he already has one), and calls FOPEN for the remote file.

With this method, Compaq easily can distribute the ASK MANMAN package across several machines, with *no changes to the application.* Karl advises that there be a logical split in the application, where files may be moved. In the case of MANMAN, the three major components are purchasing, manufacturing and physical inventory. Users log on to the machine that contains the component they're interested in. This ensures that most of the database access is local, with only occasional access to files on the other systems.

■ *The Inside Details.*

The software isn't terribly tricky after all. The normal FOPEN is renamed to be HP'FOPEN, and the Softsmith FOPEN routine is added to the system SL. When FOPEN is called, this routine determines which system the requested file resides on. If it's on another system, it just inserts the DS machine name into the device parameter, then calls HP'FOPEN. Nothing to it. If necessary, it opens a DSLINE and does a remote hello onto the other machine.

In UB-Delta-1 the remote logon can be done automatically by NS as part of the DSLINE command, making Karl's routine even more vanilla. There still will be a remote CI process. All that is saved is the trouble of having to do the remote hello and remote bye.

Another advantage is that this new feature takes one less NS socket.

■ *Reflecting Mirror Images.*

Miles Gilbert was designing a new Accounts Receivable system in Transact for Dexter Shoes. Unfortunately, the people responsible for names and addresses were in Boston and the people responsible for transactions were in Maine. How could Miles put the data near the responsible users when both groups needed access to all of the data?

First, Miles split the database in two: names/addresses versus transaction. Then he put a Series 40 in each location, with both databases on both machines. The users in Boston maintain the name/address database and have read access to a copy of the transaction database. In Maine, the users maintain the transaction database and have read access to a copy of the Boston database. Each site has a mirror copy of the other's database.

To keep the mirror databases in sync, Miles runs Silhouette in both directions between the sites. Silhouette transmits name/address changes from Boston to Maine, where it applies them to a mirror copy of the database. This keeps Maine updated to within a few minutes of real-time.

Silhouette also transmits transaction changes from Maine back to Boston, where they are updated to another mirror database. Each site has all the current information, has control of its own data and provides emergency backup for the other site.

### Put Heavy CPU Work On PCs

Applications such as spreadsheets, graphics and word processing are notorious consumers of CPU time. These benefit from being on their own dedicated computers. PCs are a good choice, because a dedicated PC often performs better than a busy Series 70 on CPU-intensive applications.

Word processing is another application that definitely should be on a PC. If you're running HPWORD or some other word processing package on your HP 3000, you're paying dearly for it. You should not allow any word processing on your HP 3000 unless it's dedicated to word processing. HP has been advocating this approach for a few years, and the development thrust of its software has been in this direction, with more PC-based software and access software to upload and download the data. You don't *necessarily* need the latest and greatest integrated software for uploading and downloading. But Reflection from Walker Richer & Quinn, Inc. (Seattle, WA) will do the job well.

■ *Integrating PCs And 3000s.*
Rolf Schleicher in Hamburg controls a network of over 200 PCs from his HP 3000, using Reflection and a few other tools. The 3000 automatically logs onto the PCs at night and backs up the hard disks for the users.

He also updates the dBASE files on the PCs nightly with the latest information extracted from corporate IMAGE databases. He often uses his larger PCs as attached processors to expand the power of his 3000 (think of how many MIPS there are lying around unused at night!).

For example, if Rolf has a large statistical analysis to do, he may download the data to a PC, start the analysis running and upload a single number later as the result. He finds that many system management tasks are easier to program in LOTUS than in COBOL; for example, managing disk space and file use by doing :LISTF and :REPORT into a LOTUS spreadsheet for analysis.

■ *A Company That Is Heavily Invested In PCs.*
The folks at Compaq Computers have some 900 PCs. Instead of downloading raw data files from the HP 3000, they have summary files lying around that they download using Reflection, feeding them into LOTUS or graphics or whatever. They do all their graphics on the PCs except for one giant run of 85 graphs in DSG, which comes at the end of the month on the laser printer. It ties up an entire Series 70 until it's over. They don't attempt to do anything else on that machine until the graphs are printed. But all of their other graphics, what-if graphics, presentation graphics, are done on the PCs. This keeps the graphics hogs off the HP 3000s.

### Upgrade To Faster Hardware

■ *The Series 70 Is A Winner.*
People we talk to say their Series 70s are terrific, especially when they're loaded up with eight or nine megabytes of memory. They have a lot more horsepower than a Series 68.

If you're on a Series 68 or smaller, you might consider going to a 70 instead of a 930 or 950. The Series 70 is so much more powerful than a 68 that we've heard it's impacting the market for the Series 930.

When Longs Drugs upgraded one of its Series 68s to a Series 70, it went to U-MIT, TurboIMAGE, and converted from Desk III to Desk IV all at the same time. At first the users didn't see any difference in performance. But then they discovered that Desk IV ran 40 percent slower than Desk III! When they fell back to Desk III, the system really took off! The extra power of the Series 70 masked the poor performance of Desk IV. Now that's horsepower, to be able to swallow up application problems as easily as that.

When the folks at Boeing upgraded their TMS manufacturing machine from a 68 to a 70, they noticed a tremendous improvement in performance. Their two-day backlog of batch jobs disappeared!

■ *Use LAN/3000 Instead Of DS.*
A LAN will not reduce your system load, but users report that it offers much higher throughput than DS with just about the same overhead. You have to replace an INP with a LANIC, and string coaxial cable instead of regular wires.

Bill Gates at Longs Drugs says that for the small price of three percent more CPU, a job that was taking 50 minutes over a 56-KB line using DS now takes eight minutes over the LAN.

When Northern Telecom (Lachine, Quebec, Canada) went from DS to LAN, it got more communication throughput without noticeable increase of CPU overhead. There are three Series 70s, a Series 852, and two Series 90002 connected together in the same room. Besides being faster than DS, it costs less, because only one LANIC per machine instead of many INPs is required. Using a "vampire tap," another

> *People we talk to say their Series 70s are terrific, especially when they're loaded up with eight or nine megabytes of memory.*

machine can be added to an active communications wire without affecting any other machines.

7933XP drives with hardware cache seldom help and actually hurt performance. A few sites have reported improved performance using 7933XP drives in place of MPE caching, but many more have not.

Perhaps the Eagle XP drives will work better. They have 2 MB of cache space, are 20 percent faster and have reduced the "pep" overhead to one millisecond per access (from six to 10 ms.).

The new Falcon drives from EMC Corporation (Hopkinton, MA) also show promise. They have 4 MB of cache, incorporate faster drives than the Eagles, provide caching for requests of up to 32 KB (instead of four KB) and claim to have a smarter cache algorithm.

More memory can help, unless you already have three megs for caching. New CRT's for Forms Cache (2394) can improve response time.

New 9600-baud modems can make remote users smile. The Microcom (Norwood, MA) AX/9624c modems understand HP's Enq/Ack protocol and have worked well on our Series 37 at Robelle Consulting (Langley, B.C., Canada).

Remember, response time is perceived by the user, and a large part of that perception is not the processing efficiency of the programs, but the speed of the datacomm gear.

## Review Batch Processing

■ *Successful Sites Discourage Online Reports.*
If you allow users unlimited access to run reports on the production machine, why should we feel sorry for you? You're getting the slow response that you asked for. Reports should run in batch, because that's where you can control the total number of any time.

■ *Concurrent Batch Jobs.*
On a Series 70, there's enough extra power to allow concurrent batch jobs. Some sites allow six or eight executing batch jobs at the same time. Bose and Turbo Resources both restrict concurrent batch jobs to different user.accounts. See MBQ in the Contributed Library for ideas on how to control this.

■ *The Night Time Is The Right Time.*
To ensure good response for online users, most of the successful sites we contacted had a policy of controlling the number of concurrent batch tasks allowed during the day. The 3000 will run just fine all night long, without anyone watching it. Many shops are shifting work from prime shift to graveyard.

■ *What To Do When Overnight Jobs Don't Finish?*
"Unfinished nightly jobs" is now a common complaint at HP shops, especially at the end of the month (perhaps because people listened to advice to shift work to the evening hours).

In our survey, we heard several methods for improving batch throughput: Upgrade to a Series 70, get a separate CPU for reports, require department-head approval on job requests, reduce backup time, increase block sizes and, the most successful strategy, apply MR NOBUF tools wherever possible (As an HP SE said, "I have seen incredible speed improvement from front-ending QUIZ with SUPRTOOL. Software solutions to performance problems often show gains of 10 or 20 times. Hardware solutions, with no improvement in the efficiency of the underlying software, usually show gains of less than one or two times.").

■ *Backup Taking Too Long.*
Many people are spending two to four hours per night on backup. If you run out of night, there are ways to reduce backup time. Get high-speed tape drives. Look at BackPack from Tymlabs Corp. (Austin, TX). HP's Copycat program and the FCOPY-FAST option of MPEX will do a high-speed disk-to-disk backup, after which you can let the users and jobs on again and do disk-to-tape backup at your leisure. Elbert Silbaugh at Boeing uses this method and keeps his system available 23.5 hours a day.

Another Boeing site in our survey wrote a privileged program to copy the database disk-to-disk *while the users are still accessing it in read-only mode.* Their system is available 24 hours a day. (Adager also can copy a database while it's open for read-only.)

## Use MR NOBUF Tools And Optimum Block Sizes

Problem: *One of the most common destroyers of system performance is the notorious serial scan. When you copy an enormous file, reorganize a KSAM file or select 100 records to report with QUIZ by reading every entry in a million-record dataset, you're bogging down the computer. The default methods of doing a serial scan are extremely inefficient on the HP 3000.*

Solution: *One of the most impressive ways to speed up serial I/O is to use MR NOBUF (multi-record non-buffered, not Mister Nobuff). You can write your own code to take advantage of MR NOBUF access if you're careful, but you don't need to. You can purchase tools that do it for you.*

Popular tools that use MR NOBUF access are HP's DSCOPY (you can use DSCOPY for copying files to the same system), HP's COPYCAT for file copying and backup, MPEX's FCOPY/FAST and Tymlabs' COPYRITE for file copying and duplication (powerful for KSAM users).

Robelle's SUPRTOOL does MR NOBUF serial file access for IMAGE datasets (and any other file type) and Running-Mate replaces serial dataset reads in applications.

■ *The Power Of MR NOBUF.*
We got a call a while ago from a fellow who didn't even know he had SUPRTOOL on his system, because it came bundled with another package he had bought. He found it and the documentation on his system, so he started using it.

He had a QUIZ job that normally took two hours to run, cruising through a huge database. A total novice, using

the instructions in the manual, he used SUPRTOOL to front-end his QUIZ report. The total time for his daily job went from two hours down to 15 minutes.

One of the shops we interviewed still uses a service bureau for some big accounting merges in IBM batch. They're considering that if the Spectrum is big enough, they might use it for that. They used to have four service bureaus. Now they're down to one. They brought things in-house by giving them their own machines, finding packages like mailing-list software front-ended by SUPRTOOL.

The folks at Turbo Resources used their HP 3000 to bill their credit card customers. At the end of the month, they had a batch program that generated one million disk I/Os reading a 90-record control file. Sixty of the records were unnecessary and, after reblocking the file, they were able to read it in one disk I/O. They now keep the control information in a table in memory, reducing one million disk I/Os to one.

■ *Block Sizes.*

The default block factors (number of records per physical disk block) are usually wrong. For big batch disk files, the maximum block size is now about 14K words (REC = 14336), while the default still is the smallest block that will fit. The bigger the block, the faster the programs will run. For IMAGE databases, the default block size is 512 words, as it has been since 1975. Many people we contacted in our survey were using 1024 words or more.

## Compile Your Fourth-Generation Applications

Problem: *Interpreted Transact and other 4GLs consume too much CPU time.*

Solution: *Compile Transact source using the Fastran compiler from Performance Software Group (Sandy Spring, MD).*

When Cathy Vanderburgh was at Macmillan Blodel, she wrote up her experiences with Fastran as *Riding Herd on a CPU Hog*: "We recently developed a Transact system that included a large (15,000 lines) and complex (10 screens) data-entry program. After installation, the response times for the program varied from slow when the machine (an HP 3000/64 with MPE IV) was lightly loaded to abysmal when the machine was heavily in use. Yet none of the other users on the system were experiencing similar problems at any time.

"We ran OPT/3000 to observe the execution of the program. The CPU time needed to interpret the IP code plus the complexity of the program was causing the MPE schedule to class the process as a 'CPU hog' and to penalize it by dropping its execution priority. The only way to improve the response time would be to reduce the excessive CPU use.

"Fortunately, this story has a happy ending. We discovered a piece of software called Fastran that compiles Transact source code into an executable program. On evaluation, we found that a Fastran version of the program used ¼ to ⅓ of the CPU of the original Transact program, enough of a drop to bring the response back to an acceptable level. The user now enjoys(?) the same response patterns as everyone else on the machine. And the moral of the story? Without Fastran, of course, the author of the original program now would be busily rewriting it in COBOL. Plenty of programmers have discovered the hard way the functional limits of tools like Transact."

At Canadian National Railways (CNR), where a large online application is written in Transact, compiling the application with Fastran (Performance Software Group, Sandy Spring, MD) led to a CPU reduction of over 60 percent and a stack size reduction of 25 percent. Single-user elapsed run times did not improve much, but as more users were added, the reduced CPU requirements produced shorter elapsed run time. These numbers are for an I/O-bound application where most of the time is spent in the database intrinsics and the file system; on CPU-intensive tasks the reduction can be considerably greater.

At Kitsap County they use Fastran over Transact wherever possible, because the programs run faster. However, they've found a few cases that Fastran can't handle. If a program needs extensive table handling, they choose COBOL over Transact.

Dexter Shoes was described earlier as running large manufacturing and distribution operation on a network of six Series 40s. The entire application there was coded from scratch in Transact. The folks there report that this gives them the ability to respond to user suggestions in days instead of months. The reason they can get away with only Series 40s, instead of Series 68s or 70s, is that they compile the programs with Fastran.

Larry Kemp of HP Bellevue has found Fastran about 25 percent slower than COBOL and 50 to 98 percent faster than Transact (an eight-hour job reduced to eight minutes was the best he ever saw!). An alternative 4GL that he found to give excellent performance is Protos (Protos Software Company, Austin, TX); it generates a COBOL program for execution. And, finally, no one says you can't rewrite your most frequently used program in COBOL (use system logging to find out which program it is.)

## Get OMNIDEX For Fast Online Database Searching

IMAGE provides calculated read, chained read and serial read. OMNIDEX from Dynamic Information Systems Corporation (DISC, Denver, CO) adds record selection across multiple fields, generic retrieval and sorted sequential access, multiple keys in masters and keyword retrieval on text data. It does this by adding another structure to IMAGE's: the binary tree. Traversing this tree is fast, fast, fast.

At HP they use OMNIDEX in the Response Center to index bug reports. That's how they can find out instantly who else has had a system failure 916 on Series 37 under T-MIT with a full moon.

OMNIDEX indexes every word, not just the manually-assigned "keywords" as in the old SSB system. Doug Iles of HP says, "We could enter partial values and/or full values from several different fields and find five qualifying records out of 50,000 in seconds."

The people at DISC distinguish between "informational" data — data that you want on the system for doing inquiries — and "operational" data — data generated by the transactions of the organization.

For example, in an order-processing system, active orders are operational; customer and vendor master records are informational. Operational data is volatile and lightly indexed. Informational data is static and can afford to be highly indexed for fast, low-cost retrieval. In a general ledger system, the transaction dataset is operational. You do data entry and editing with it. When the transaction is completed, you post it to the ledger dataset, where it becomes informational data. You no longer modify it (much), but you need to ask numerous complex questions about it. OMNIDEX gives you the ability to index everything in your information data. You can use batch time to update the indeces, instead of online time.

Users also apply OMNIDEX to replace KSAM. The index-sequential part of OMNIDEX (called IMSAM) will reindex about one million keys per hour on a Series 70 (versus 20 to 30 hours with KSAM).

For example, Kim Everingham at Consolidated Capital reports that they use OMNIDEX extensively in their tracking system for investors and investments. The power of OMNIDEX indexing allows their official IMAGE structure to be very simple: masters for entities and details for transactions.

They have 4.5 million sectors of data, 250 QUICK screens, 12-15 databases and 35-40 users on a Series 70. Without OMNIDEX, the application would require an IBM mainframe. Within one or two seconds they can identify an investor and the investments he is involved with, even if the investor only gives a vague or partial description of himself (e.g., trust company, Ralph, Minneapolis).

They do all updates online, including updates of the OMNIDEX indices. The only exception is the entry of new investments — that's done in a nightly batch job due to the serious impact on response.

They have plenty of horsepower with the Series 70; the only bottleneck is that QUICK consumes about 60 percent of the CPU time, but this hasn't impacted response time yet. They also use SUPRTOOL for ad hoc extracts and as a QUIZ front-end.

Kitsap County Government is an HP site that gets a lot of work done without hitting the limits of the HP 3000 line. Jim Kellam, the manager, started with a Series 48, overloaded it, then added a Series 68 and left the 48 for development. He reports that OMNIDEX inquiries are unbelievably fast ("find all the voters named Smith" instantly replies "1200 entries found"), but can be abused, just like any tool.

For example, one of their programs opens all eight databases at the start, in case you might need then. Installing OMNIDEX implies an extra open and another extra data segment, the equivalent of 16 DBOPENs per user. The users sometimes get in and out of the application to access other software, so they pay this startup overhead more than once per day.

The IMSAM part of OMNIDEX allows you to define concatenated keys with pieces from three different datasets. Jim feels that they may have overused these features, because he observes slow response with some of these bizarre keys. —*Michael Shumko handles Technical Support, Robert Green is president and David Greer is senior programmer at Robelle Consulting Ltd., Langley, B.C., Canada.*

Would you like to continue to see articles on this topic?
Enter On Reader Card
yes 806 no 807

## PSP/Plus Adds Versatility To LaserJet

With PSP/Plus from OPT, all the versatility of laser printer technology, previously available only on the HP 2680 and HP 2688, is now available with HP's LaserJet family of printers.

Until now, the LaserJet was unavailable to TDP users, and its graphics capabilities have been unavailable to HPWord users. With PSP/Plus, you can take full advantage of the formatting power of TDP, including environment files, embedded graphics, forms overlay, up to 31 fonts per page, logos and more.

HPWord users also have access to graphics, downloadable fonts and forms. And, when using PSP/Plus, the LaserJet is system-spooled so HPWord and non-WP users can share the same printer.

HPDraw, EZChart, and DSG files can be plotted directly to the LaserJet or, if you prefer, printed as a figure file. PSP/Plus provides the HP 3000 users full access to IFS/3000 and IDSForm. Either HP-supplied or custom environment files created for the HP 2688 may be used. There is an option that allows you to use the LaserJet's soft fonts as HP 2688 character sets.

PSP/Plus is transparent to users. Once the product has been installed, output can be directed to any of HP's laser printers, including the new LaserJet Series II and LaserJet 2000.

For more information, contact OPT at 299 W. Foothill Blvd., Suite 230, Upland, CA 91786; (800) 858-4507.

**Enter 907 on reader card**

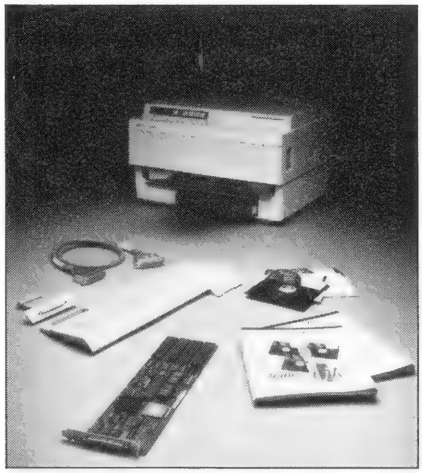
## Imagen Introduces PC Publisher Kit

Imagen Corporation has introduced the DDL-based PC Publisher Kit consisting of a raster image processor on a card that resides in an IBM PC or compatible.

The PC Publisher Kit allows the existing base of HP LaserJet and other Canon CX-based laser printers to be upgraded to provide full desktop publishing features including full-page graphics and powerful font capabilities.

Hewlett-Packard will publicize and promote the PC Publisher Kit for LaserJet as part of its overall office printing solutions strategy.

The PC Publisher Kit, priced under $2,000, consists of a raster image processor board that connects to the print engine via a high speed video interface. Software is provided to upgrade the LaserJet to the full power of DDL, while maintaining compatibility with HP's Printer Command Language (PCL) files.



***The PC Publisher kit allows the HP LaserJet and other Canon CX-based laser printers to be upgraded to provide full desktop publishing features.***

A special utility package included provides support for virtually any existing PC application software package. This capability allows existing application files to be printed using the PC Publisher Kit by emulating printers already supported by those applications. This utility, DDL LaserControl, was developed for Imagen by Insight Development Corporation.

Desktop Publisher and CAD/CAE application software support of the PC Publisher Kit will be available for leading applications packages.

Imagen is headquartered at 2650 San Thomas Expressway, Santa Clara, CA 95052-8101. Call (408) 986-9400 for more information.

**Enter 908 on reader card**

## MJH Systems Ships Pascal/Modcal Formatter

MJH Systems has announced shipments of a major new release in its source code formatting line of products.

Release 2.00 of the PASFORM Pascal source code formatter provides at least twice the power and flexibility of its predecessor by providing a host of new formatting options. Additionally, a companion formatter for the Modcal language also has been developed. This new product, called MODFORM, has been designed with the same interface and design capabilities found in the PASFORM formatter and conforms to the full August 1985 specification of the HP Modcal language.

Release 2.00 provides another enhancement in the area of capitalization and formatting capabilities which was not available before. The products have been redesigned to functionally separate the capitalization and formatting functions, thereby allowing the tool to be used as a capitalizer, formatter or both for all (or selected pieces) of the code. This capability allows programmers with artistically formatted sections of code to maintain their formatting layout and still reap the benefits of having the code tokens consistently capitalized.

PASFORM is available for the full line of HP computers — HP 9000 Series 200/300 Pascal Workstation and HP-UX, HP 1000 RTE and HP 3000 MPE. MODFORM runs on the HP 3000 MPE and HP 9000 Series 840 HP-UX systems.

MJH Systems is located at 9375 Albany, Boise, ID 83704; (208) 327-0011.

**Enter 925 on reader card**

## Universal File Manager Improves Connectivity

Net/One Universal File Manager from Ungermann-Bass, Inc., is the first product to provide universal file access and management between any combination of IBM and Apple Macintosh personal computers and Hewlett-Packard, IBM and Digital Equipment Corp. mainframes and minicomputers.

The software enables files to be transferred between different types of computers and transformed into different formats and provides security by allowing data access only to authorized users.

With Universal File Manager, a networked PC user who is familiar with host commands can perform a directory search on an IBM host, obtain data from a file residing on that host, combine the data with information residing in a PC-based spreadsheet and transfer the resulting file to a VAX departmental computer.

Developed by Linkware Corporation (Waltham, MA), a wholly owned Ungermann-Bass subsidiary, the Universal File Manager consists of a core product called the Universal File Transfer, and two options, the Universal File Transformer and the Universal File Administrator.

Ungermann-Bass, Inc., is located at 3900 Freedom Circle, Santa Clara, CA 95052-8030. Call (408) 496-0111 for more information.

**Enter 910 on reader card**

## Simpleplot Provides omniBUSS Driver

Bradford University Software Services Ltd. (BUSS) has developed an extension to the Simpleplot library to provide a standard graphics interface, the omniBUSS driver.

The omniBUSS driver is capable of driving any combination of the following generic device types: Devices using Tektronix 4010/4014 code, Tektronix 4100 series code, Hewlett-Packard HP-GL code (e.g., HP7475, LVP16), DEC ReGIS code, or DEC SIXEL code.

The omniBUSS driver is configurable to address a specific set of devices at a given site and can be extended as new equipment becomes available. A program using the one device driver, therefore, can produce graphics output on an open-ended set of terminals without recompilation or relinking.

BUSS Ltd. is located at 29 Campus Road, Bradford, West Yorkshire BD7 1HR.

**Enter 911 on reader card**

## Acer Introduces HP-Compatible Printer

Acer Technologies Corp., a newly-formed affiliate of Multitech Electronics Inc. (MEI), has introduced a small-footprint, low-cost, desktop laser printer, which provides 1.5 MB of memory for a full page of high-resolution graphics.

With a six-page-per-minute text resolution print speed, the new Acer LP-75 Laser Printer is fully compatible with the HP LaserJet Series II, with 300 X 300 dots per inch and access to the more than 500 software programs designed for HP printers.

The printer comes standard with nine resident fonts. In addition to its HP LaserJet Series II compatibility, the Acer LP-75 has access to a whole range of additional cartridge fonts as well as software fonts that can be downloaded to the printer.

Up to 500 KB of RAM can be added for a total of 2 MB of memory, providing an even larger page buffer with additional room for downloading software fonts. In all, more than 16 fonts may be used per page for creative desktop publishing.

The LP-75, priced at $2,995, incorporates the Ricoh six-page-per-minute engine and weighs 37.5 lbs.

Multitech Electronics Inc. is located at 401 Charcot Avenue, San Jose, CA 95131. Call (408) 922-0333 for more information.

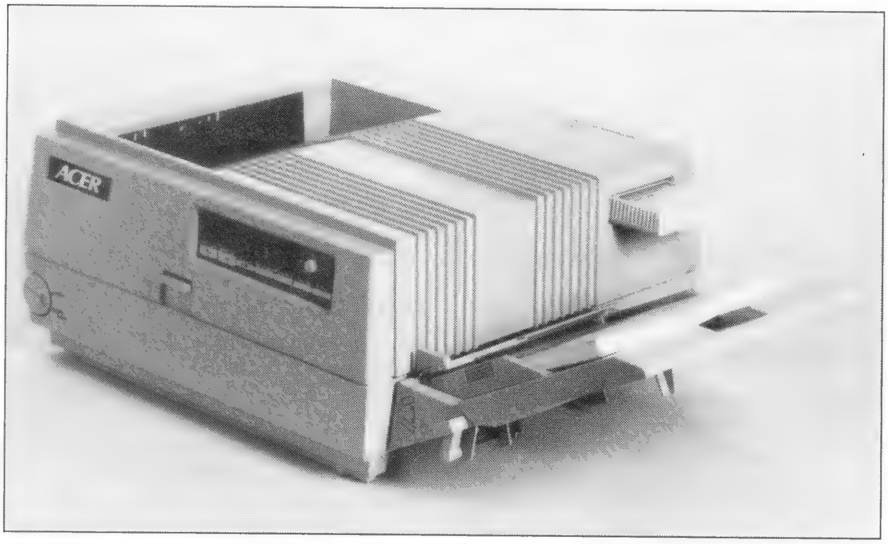
**Enter 909 on reader card**

## Harris Develops DISOSS For 9300

Harris Corporation's National Accounts Division has developed a Distributed Office Systems Support (DISOSS) gateway for the Harris 9300 network communications system.

The DISOSS gateway, priced at $2,995, allows all workstation users participating on a Harris 9300-based electronic mail system to interchange documents, messages and files through IBM's DISOSS host program.

By sending indirectly through DISOSS, users of the Harris electronic mail system can send and receive all varieties of mail with




*Acer Technologies' LP-75 Laser Printer.*

users of DISOSS-compatible systems such as the Hewlett-Packard 3000, DEC VAX and IBM System 36.

Harris Corporation's National Accounts Division is located at 16001 Dallas Parkway, Dallas, TX 75248. Call (214) 386-2000 for more information.

**Enter 915 on reader card**

## EEsof Releases Microwave SPICE 1.1

Microwave SPICE 1.1, the latest version of EEsof's non-linear microwave/RF circuit design and simulation tool, features a new signal-power analysis component, Fast Fourier Transform (FFT) capability, extended user-definable functions for independent sources, and more.

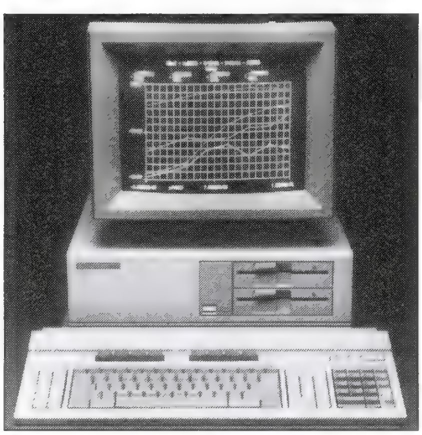
This version of mwSPICE makes non-linear simulation easier, shortens the design cycle and saves time for microwave and RF engineers in the lab.

A new signal-power analysis capability features two additional elements, power source and power meter, as well as a new signal-power analysis command. With these three additions, mwSPICE users now can display complete signal power at specific frequencies, sweep power levels and frequencies over specified ranges, compute relative signal strengths, and use multiple input signals to generate and compute intermodulation distortion.

Microwave SPICE 1.1 runs on the Hewlett-Packard Vectra PC, IBM PC-XT and PC-AT and COMPAQ DESKPRO 386. It will be available for the HP 300 series and other systems later this year.

Contact EEsof, Inc., at 31194 La Baya Drive, Westlake Village, California 91362; (818) 991-7530.

**Enter 914 on reader card**



*Microwave SPICE 1.1 from EEsof, Inc.*

## MOREHELP For 3000s

Apogee, a consulting and custom software house, announces its first general market product for HP 3000 users, MOREHELP.

Expanding on MPE'S :HELP subsystem, MOREHELP is accessible from the Command Interpreter, from User Defined Commands, or from within a user's program.

Just as MPE :HELP provides information on MPE commands, MOREHELP provides information on additional aspects of the HP 3000. MOREHELP is available on system intrinsics including parameters and options; Image databases and intrinsics; QUERY; VPLUS including intrinsics; scientific, mathematical and compiler libraries; SORT/MERGE; FCOPY; DEBUG; file formats; character codes; system management and more.

MOREHELP is available on 1600 BPI magnetic tape ($195-$295), in manual format (printed copies), and on microfiche. It is MPE MIT independent and can be installed on any HP 3000 running any version of MPE-IV, MPE-V or MPE-XL.

Apogee is located at 4632 West Frankfort Drive, Rockville, MD 20853; (301) 460-0021.

**Enter 919 on reader card**

## KLA/Express 3000 Generates More Throughput

KLA & Associates' KLA/Express-3000 is a background process that works with the MPE Scheduler/Dispatcher to generate more throughput.

KLA/Express-3000 will give your computer the intelligence necessary to create the most efficient environment possible based on your company's needs. By dynamically raising and lowering executing priorities and controlling program execution, it easily can double or triple your system's throughput.

Contact KLA & Associates, Inc., P.O. Box 14854, Clearwater, FL 34279-4854; (813) 784-5976.

**Enter 921 on reader card**

## Wollongong Ships WIN/H9000 Software

The Wollongong Group has begun shipping its new WINS communications software for the HP 9000 Series 500 technical computers.

The product provides complete TCP/IP networking services to users of HP's 9000 Series 500 computers under the HP-UX operating system. It works in conjunction with a LAN/500 Link and, through Wollongong's TCP/IP implementation, gives HP users the ability to participate in a multi-vendor local area network for the first time.

All WINS products have the following capabilities: file transfer, network virtual terminal, electronic mail and the ability to develop custom networking applications using the WINS function libraries.

WIN/H9000, priced at $7,500, also provides a programmatic interface to allow network-oriented programs to be implemented using the "socket" interface, and supports remote commands from foreign UNIX host systems.

Wollongong is headquartered at 1129 San Antonio Road, Palo Alto, CA 94303; (415) 962-7200.

**Enter 920 on reader card**

## EMC Announces Memory For Micro 3000XE

EMC Corporation has introduced memory for Hewlett-Packard Micro 3000XE computer systems, HP's entry-level business machines.

EMC now offers 1-MB, 2-MB, 3-MB and 4-MB memory arrays for the Micro 3000XE, which has a maximum main memory capacity of 8 MB. EMC is the only company to offer this wide range of memory capacities for these computers.

EMC's Micro 3000XE memory is 100 percent compatible with the MPE operating system and requires no hardware modification. It also allows the user to take full advantage of the system's interleaving capability. It features LED activity indicators to monitor board level activity, on-board error check and correction (ECC) capability and an online/offline comfort switch.

Contact EMC Corporation, Hopkinton, MA 01748-9103; (800) 222-EMC2; in MA, (617) 435-2541.

**Enter 918 on reader card**

## C/COMPARE-HARMONIZER For Easy Upgrade

A new software utility from Aldon Computer Group will automate the process of moving to a new release of packaged software.

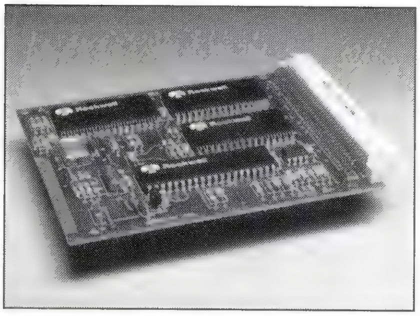
Using just one command, S/COMPARE-HARMONIZER will compare the original version of a software package, the user's modified version and the vendor's new release. Then HARMONIZER will produce

a report identifying all the added and deleted code, and it will create a compilable output file integrating all versions.

HARMONIZER allows the users to feel free to customize packaged software knowing they have a method for integrating their changes into new releases.

Contact Aldon Computer Group, Financial Center Building, 405 14th Street, Oakland, CA 94612; (415) 839-3535.

**Enter 926 on reader card**



***Rockwell's R208/201 automatically determines whether a remote modem is operating in Bell 208A/B or 201-C appropriate format.***

## Rockwell Modem Switches Between Bell Modes

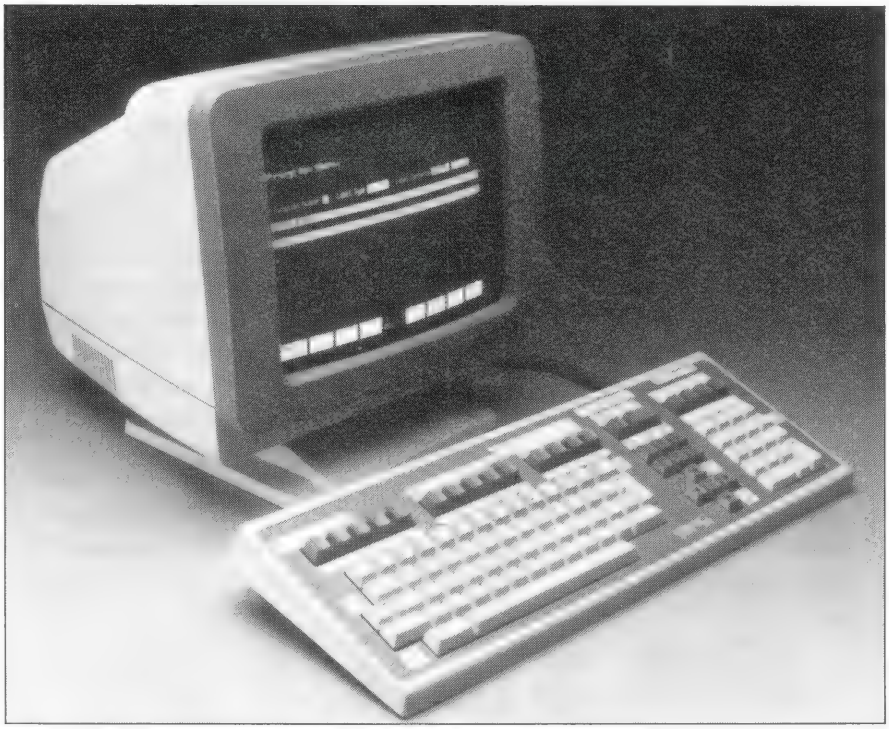
The Semiconductor Products Division of Rockwell International Corporation now offers the R208/201, the first synchronous 4800 and 2400 bps OEM modem that automatically can determine whether a remote modem is operating in Bell 208A/B or 201-C mode and instantly switch to the appropriate format.

Designed for half-duplex communication over the public switched telephone network (PSTN), as well as full-duplex over four-wire leased lines, the R208/201 satisfies the Bell 208A/B, Bell 201-C and CCITT V.27 and V.26 standards.

Key features include automatic adaptive equalization to continually compensate for line distortions; a parallel microprocessor bus interface to allow the host CPU to monitor and control features such as self-diagnostics, a CCITT V.24/RS-232-C port, dynamic receive range of − 43 to 0 dBm, and TTL and CMOS compatibility.

The Semiconductor Products Division is headquartered at 4311 Jamboree Road, P.O. Box C, Newport Beach, CA 92658-8902.

**Enter 916 on reader card**



*Teleray's new Model 20-DHP, an alternative to HP's 2392A.*

## New Features Enhance HP-Compatible Teleray

The Model 20-DHP, Teleray's alternative to Hewlett-Packard's 2392A, has been modified to incorporate a number of significant operating enhancements including typeahead and Return = Enter features and a "dual port option" that allows the terminal to independently operate and maintain sessions with two HP hosts or an HP and a DEC, or even two DECs, and which lets a user switch from one host to the other with a single keystroke.

The typeahead feature allows data (up to 256 characters) to be keyed into a buffer during the time the computer is processing a previous entry. That data then automatically is transmitted in response to the computer's next prompt.

Other 20-DHP enhancements include the addition of a soft white 14-inch CRT, to go with the amber and green choices already offered, and a simplified feature lock capability that's implemented with a single keystroke in setup mode.

The new 20-DHP, priced at $1,295, retains the original unit's 2392A emulation capability and all its other standard features. Contact Teleray, Box 24064, Minneapolis, MN 55424; (612) 941-3300.

**Enter 922 on reader card**

## Martinsound Offers MS-350+ Megamemory

Martinsound Technologies now provides a method of expanding the new HP Series 350 computer from 8 MB to a full 16 MB of online RAM memory. This expansion maintains the quality standards of the HP 350.

Martinsound exchanges HP 98258B 4-MB RAM add-on boards and adds 8 MB of RAM memory (equivalent to the HP 98258C 12-MB RAM add-on). Each upgrade is completely inspected, tested dynamically and burn-in for 200 hours. Contact Martinsound Technologies, 1151 West Valley Blvd., Alhambra, CA 91803-2493; (818) 281-3555.

**Enter 924 on reader card**

## Software On ROM For Portable Plus

Personalized Software has announced the availability of selected software programs on ROM chips for users of the HP Portable Plus.

By inserting one or more ROM chips in the Portable Plus's ROM drawer, users of the selected software programs can free up memory space and simultaneously save the trouble of using a disk drive.

Programs currently available on ROM Backup include Shelp, SideWinder, The Editor, PrinterTalk and Format110, Turbo PASCAL, Typing Whiz, Best of DOS Tools and Webster's Spelling Checker.

Shelp and SideWinder are packaged together on one 64K EPROM that sells for $95. Similarly, Shelp and SideWinder are available together on a single 64K EPROM for $95.

Best of DOS Tools is available on 128K ROM for $119, and the backup for Webster's Spelling Checker, which takes up two 128K ROMs, sells for $179. A combination package called Super ROM, which includes The Editor, Shelp, SideWinder, Format110 and a mini version of PrinterTalk all on one 128K ROM, costs $149.
Contact Personalized Software, P.O. Box 869, Fairfield, IA 52556; (515) 472-6330.

**Enter 929 on reader card**

## Bar-Code Label Manager For HP 3000

Quality Consultants, Inc., has announced a new Bar-Code and Label Management System named BCL3000. Designed for the HP 3000, the product accommodates almost any labeling need by generating user-defined labels of one to 132 characters wide by one to 99 lines long. This allows printing of small labels for filing and mailing or very large labels for shipping/manufacturing applications, as well as signs or banners.

BCL3000 allows users to design a label or bar-code format and then modify the formats needed. The fully documented system has complete security allowing the system manager to password protect its functions as desired.

Prices range from $2,000 to $5,000 depending on the number of copies ordered, HP 3000 model used and whether object or source code is desired.
Contact Quality Consultants, Inc., 1775 The Exchange, Suite 380, Atlanta, GA 30339; (404) 980-1988.

**Enter 923 on reader card**

## PSA Introduces UNI-FONTS

A library of laser printer fonts known as UNI-FONTS is now available from Professional Software Associates, Inc.

The UNI-FONT library currently supports the HP LaserJet Plus and LaserJet Series II, DEC LN03 and LN03 Plus, QMS-810, Talaris-810 and Data Products LZR-1230 printers.

UNI-FONTS are used in conjunction with PSA's Font Family software, which


**T**wenty-two percent of *HP PROFESSIONAL* Readers . . .
have DEC VAXs at their site. Are you one of them?

**A**re you a DEC computer user . . .
a buyer of DEC hardware, software, peripherals?

**D**o you read *DEC PROFESSIONAL* . . .
the leading monthly magazine for DEC computing?

- It's FREE!
- It's informative.
- It's monthly.
- It's read by over 90,000 professionals like you.
- It's time you had a free subscription too!

Read *DEC PROFESSIONAL* for information on the latest hardware, software, programs, new products, industry trends, problem solving suggestions and more. Each issue has timely, relevant DEC computing information. And each issue is a valuable reference you'll want to keep on hand for months to come. (Some of our subscribers have *DEC PROFESSIONAL* "libraries.")


## Start Your Free Subscription . . .

**Call:** Margie Pitrone, *Circulation Manager*
(215) 542-7008

**Or write:** 299 on the Reader Information Card.

**and we'll send you a subscription application today!**
***It's Free!***

ENTER 299 ON READER CARD

enables users to get more out of their intelligent printers.
Contact PSA at (800) 621-0849.

**Enter 930 on reader card**

## Compact Newsletter Issued Quarterly

Compact Software announces the new *Transmission Line Newsletter*, which gives information about Compact Software programs, enhancements, applications and the newest developments in MIMIC design.

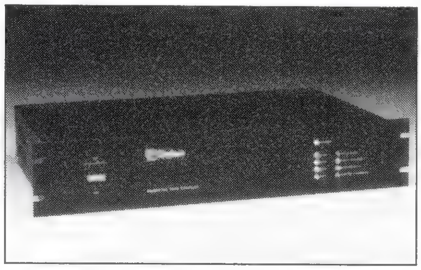
The newsletter will be issued quarterly. Contact Compact Software, 483 McLean Blvd. & 18th Avenue, Paterson, NJ 07504; (201) 881-1200.

**Enter 928 on reader card**

## 1015C Controller Allows Mag Tape Access

A newly designed magnetic tape controller that uses the standard IEEE-488 (GPIB) bus has been announced by Dylon Division of Liken Industries, Inc. The 1015C controller now allows virtually every computer user to take advantage of the ½-inch mag tape drives, including the new GGR (6250) drives.

Fully buffered up to 32 KB, the controller handles various record lengths in NRZI, PE and GCR recording modes. The system accommodates tape speeds up to 125 ips, with tape capacities up to 180 MB in GCR mode.



***Dylon 1015C, IEEE-488 (GPIB) magnetic tape controller.***

Contact Liken Industries, Inc., 130 McCormick Avenue, #112, Costa Mesa, CA 92626; (714) 540-2205.

**Enter 927 on reader card**

## Kyocera Printer Emulates HP LaserJet Plus

Kyocera Unison, Inc., has announced a new 18-page-per-minute laser printer that features 78 resident fonts (including eight foreign language character sets), seven printer emulations, four port options, 1.5-MB onboard RAM (expandable to 3.5 MB), two paper trays, full-page 300 X 300 dpi resolution and more. Its cost is under three cents per page.

The Kyocera Unison F-3010 Compact Laser Printer emulates the HP LaserJet Plus, Diablo 630, Qume Spring II, NEC Spinwriter 3550, IBM Graphics Printer, Epson FX80 or a line printer. It has parallel and serial ports (standard).

The printer comes with a page- description language, Prescribe, and four Dynamic fonts that let users create fonts from three-point to 13-inch sizes. It is available for $8,395.
Contact Kyocera Unison, Inc., 3165 Adeline Street, P.O. Box 3056, Berkeley, CA 94703; (415) 848-6680.

**Enter 931 on reader card**

## BackPack Supports Unattended Backup

The current release of BackPack, the high-speed backup facility from Tymlabs Corporation, allows users to implement an unattended backup solution, eliminating the need for a night operator. To accomplish this, BackPack stores all files in compressed format to a standard MPE disk file. This phase of the operation can be initiated at the end of the day and left to run unattended. The tape write portion of the store is deferred until the next morning when tapes are written in a high-speed background mode, concurrent with normal system use.

In addition to supporting unattended backup, this deferred approach also minimizes system downtime. Since copying files from disk to disk is much faster than copying to tape, the time during which files are unavailable for other processing is much shorter.

Version 2.21 of BackPack also provides full support for store and restore of files located on private volumes, and a special mode that delivers improved performance when backup is run in a standalone environment (no other users or jobs on the system).
Contact Denise Girard at Tymlabs Corporation, 211 East 7th Street, Austin, TX 78701; (512) 478-0611.

**Enter 932 on reader card**

## HEWLETT-PACKARD

*Buy, Sell, Rent and Repair Computers, Peripherals and Options*

★ ★ ★ ★ ★ ★

SALES (408) 270-1100 SERVICE (408) 270-1170

★ ★ ★ ★ ★ ★

FAX — (408) 270-1183
ELM — 62015192
TELEX — 9102500341

**CRISIS COMPUTER CORP.**
2298 Quimby Road
San Jose, CA 95122-1356

ENTER 186 ON READER CARD

## HP CAREERS NATIONWIDE

**SEARCHING FOR THE RIGHT OPPORTUNITY?**

Look no further! Specializing 100% in the HP marketplace, Diane Amos can end this search. With her proven success record and 12 years experience, she will lead you to the right career path. Numerous opportunities available.

Call Diane Amos, C.P.C.

Amos & Associates

633B Chapel Hill Road
Burlington, NC 27215
(919) 222-0213

ENTER 191 ON READER CARD

## BUY • SELL • RENT • LEASE

**HEWLETT-PACKARD Computer Equipment**

SERIES 80, 100, 200, 300, 9800

*ALSO . . .Plotters, Printers, Disk Drives, CPUs, Printers, etc.*

Available for Immediate Delivery

**(713) 855-4528**

**TSA**

TECHNICAL & SCIENTIFIC APPLICATION, INC.
4654 Highway 6 N., Suite 101
Houston, TX 77084

ENTER 185 ON READER CARD

## Save up to 60%!

***Top dollar when you sell!***

We buy or sell any HP equipment, including:

COMPUTER SYSTEMS • DISK DRIVES
SOFTWARE • ACCESSORIES
TERMINALS • MAG TAPES

*All Equipment Guaranteed.*

Specialists in hard-to-find equipment.
Immediate delivery of in-stock items.

*CALL Bill Alexander at*

**(312) 354-7676**

COMPUTER MEDIA, INC.
140 North LaGrange Road
LaGrange, IL 60525

ENTER 138 ON READER CARD

## HP

**9000,3000,1000,**

**100,80 series or ATE**

**CompuChange**

**813-796-8388**

**Buy • Sell • Trade**

ENTER 140 ON READER CARD

## NORCO COMPUTER SYSTEMS, INC.

*Hewlett-Packard Quality at a NorCo Price*

**1000 • 3000 • 9000 • 250**

***BUY — SELL***
***TRADE — LEASE***

Processors, Peripherals and Systems

**30311 Clemens Road**
**Cleveland, OH 44145-1018**

**216-892-1920 1-800-892-1920**
(Outside Ohio)

ENTER 184 ON READER NUMBER

# ADVERTISERS INDEX



*Continued from page 82.*

Adager
Apartado 248
Antigua, Guatemala
(502) 232-4333
**Enter 653 on reader card**

Carolian Systems
3397 American Drive, #5
Toronto, Ontario, Canada L4V 1T8
(416) 673-0400
**Enter 651 on reader card**

Cognos Corporation
2 Corporate Place I-95
Peabody, MA 01960
(617) 720-1503
**Enter 654 on reader card**

ASK Computer Systems
730 Distel Drive
Los Altos, CA 94022
(415) 969-4442
**Enter 658 on reader card**

Dynamic Information System Corp.
910 15th Street, Suite 640
Denver, CO 80202
(303) 893-0335
**Enter 666 on reader card**

EMC Corporation
Hopkinton, MA 01748-9103
(800) 222-EMC2
(617) 475-2541 in MA
**Enter 661 on reader card**

Hewlett-Packard
1830 Embarcadero Dr., Building 9C
Palo Alto, CA 94303
(415) 857-8500
**Enter 667 on reader card**

Holland House
P.O. Box 1749
Beeville, TX 78102
(512) 287-3417
**Enter 656 on reader card**

Microcom
1400 A Providence Highway
Norwood, MA 02062
(800) 822-8224
**Enter 662 on reader card**

NSD, Inc.
1670 S. Amphlett Blvd., Suite G
San Mateo, CA 94402
(415) 795-0640
**Enter 657 on reader card**

OPT Inc.
299 W. Foothill Road Blvd., Suite 230
Upland, CA 91786
(714) 593-1681
**Enter 650 on reader card**

Performance Software Group
Post Office Box 1464
Sandy Spring, MD 20860
(301) 242-6777
**Enter 664 on reader card**

Protos Software Company
American Plaza, Suite 935,
300 West 5th Street
Austin, TX 78705
(512) 480-0865
**Enter 665 on reader card**

Robelle Consulting
8648 Armstrong Rd., R.R. No. 6
Langley, B.C., Canada V3A 4P9
(604) 888-3666
**Enter 652 on reader card**

Softsmith
9301 Jones Road, Suite 132
Houston, TX 77065
(713) 890-4182
**Enter 659 on reader card**

Telamon
1615 Broadway, 11th Floor
Oakland, CA 94612
(415) 835-5603
**Enter 655 on reader card**

Tymlabs Corporation
211 East 7th Street
Austin, TX 78701
(512) 478-0611
**Enter 663 on reader card**

Walker Richer & Quinn Inc.
2825 Eastlake Avenue East
Seattle, WA 98102
(206) 324-0350
**Enter 660 on reader card**


HP Professional

**Professional Press, Inc.**
921 Bethlehem Pike
Spring House, PA 19477
(215) 542-7008

## BUSINESS OFFICE

**Associate Publisher/ Marketing & Sales** Leslie Ringe

**Sales Manager** Jeffrey Berman

**Marketing & Sales Coordinator** Cynthia Leitzel

## REGIONAL OFFICES

**NEW ENGLAND/NEW YORK/ EASTERN CANADA**
Cynthia Davis
238 Bedford Street, Suite 3
Lexington, MA 02173
(617) 861-1994

**MID-ATLANTIC/ INTERNATIONAL**
Leslie Ringe
238 Bedford Street, Suite 3
Lexington, MA 02173
(617) 861-1994

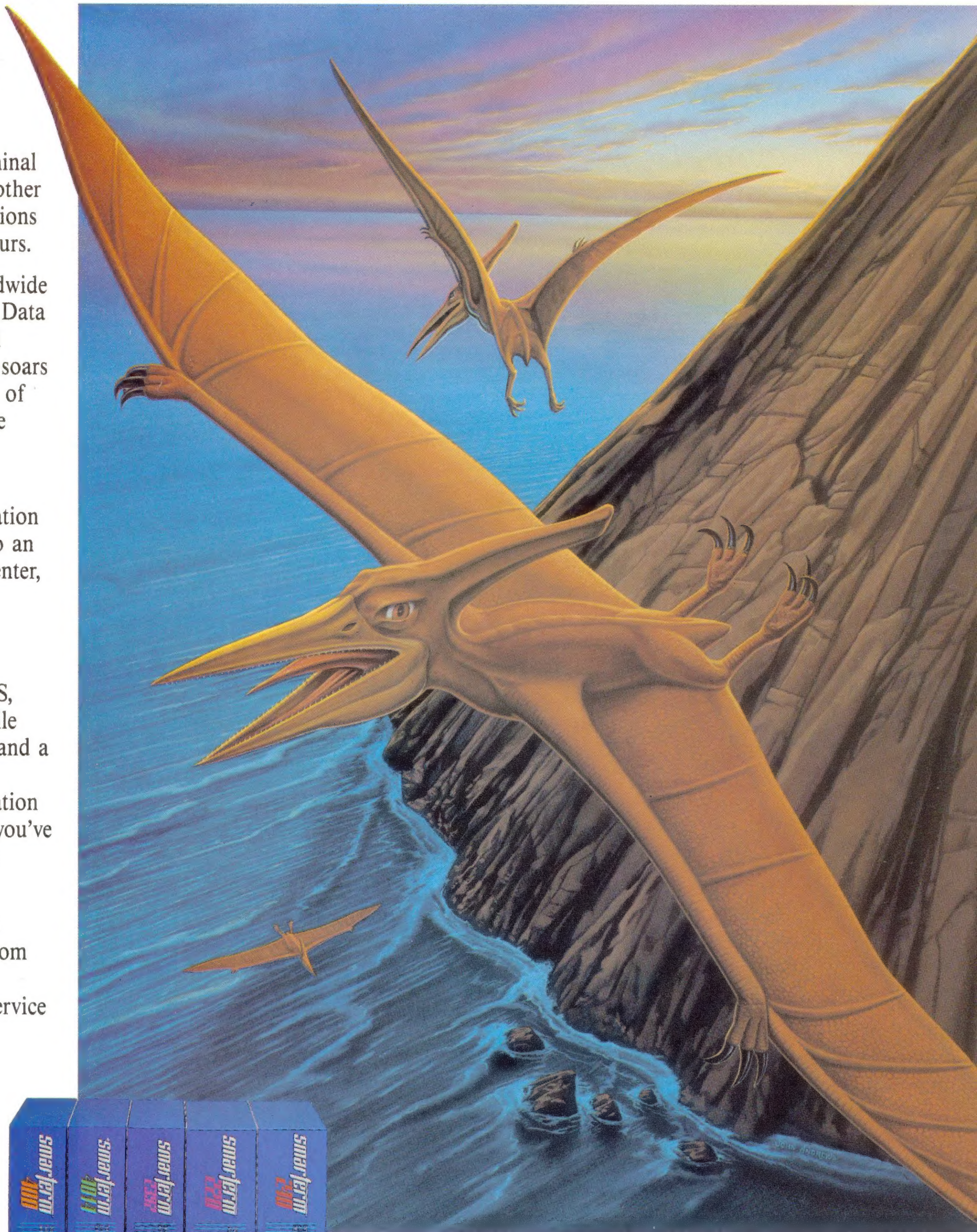
**MIDWEST/SOUTHEAST/ SOUTHWEST**
Peter Senft
921 Bethlehem Pike
Spring House, PA 19477
(215) 542-7008

**SOUTHERN CALIFORNIA**
Terry Buckley
The Buckley Companies
881 Dover Drive
Newport Beach, CA 92663
(714) 722-1242

**NORTHERN CALIFORNIA/ PACIFIC NORTHWEST**
A. G. Germano
715 El Camino Real
Suite 206
San Bruno, CA 94066
(415) 873-3368

**ONTARIO, CANADA**
Helen Marbach
921 Bethlehem Pike
Spring House, PA 19477
(215) 542-7008

# The SmarTerm® Theory of Software Evolution
# What *Really* Happened to the Dinosaurs?



## *T*hey were replaced with advanced new species.

After you've tried SmarTerm terminal emulation software, you'll think other PC-to-minicomputer communications links are as advanced as ...dinosaurs.

SmarTerm already is known worldwide as the leader in advanced DEC*, Data General* and Tektronix* terminal emulation software. Now Persoft soars to greater heights with emulation of another popular text terminal, the Hewlett-Packard* HP 2392.

SmarTerm 2392 offers you exact feature-for-feature terminal emulation and more. It connects your PC to an HP3000 Personal Productivity Center, another PC, and services like CompuServe* or Dow Jones News/Retrieval*.

Easy to use setup, Branch to DOS, Kermit and Xmodem error free file transfer, programmable softkeys, and a simple script language give you complete control. Add our dedication to quality technical support and you've got one "smart" communications software buy!

The SmarTerm family of terminal emulation products is available from your local dealer. Request more information through the reader service number below. Or contact:

Persoft, Inc.
465 Science Drive
Madison, WI 53711 USA
Phone (608) 273-6000
FAX (608) 273-8227
TELEX 759491


persoft®

ENTER 116 ON READER CARD

**SmarTerm Terminal Emulation Software**
**. . . The Natural Selection**

# What keeps Reflection top of the line...

# ...is our bottom line.

# SUPPORT

**is as much a part of a terminal emulation package as the software itself.**

That's why when you buy Reflection, you get **expert telephone support. Free of charge. Forever.**

There's extensive documentation, unlimited telephone support and an electronic bulletin board. Should a problem arise, a technical specialist will help you quickly trouble-shoot and resolve the matter.

The accuracy of Reflection's terminal emulation is legendary. Advanced features include command language, Local Area Network support, background processing, keyboard remapping, micro to host back-up, and several error-checking file transfer protocols. These features let your personal computer and host work as a team.

Reflection is available to connect a wide range of personal computers to Hewlett-Packard and Digital hosts. Call today for the connection that's right for you.

Your bottom line will reflect the results.

**REFLECTION™**
SERIES
SOFTWARE

ENTER 180 ON READER CARD

from Walker Richer & Quinn, Inc.
2825 Eastlake Avenue East, Seattle, WA 98102
**1-800-872-2829**